# Evaluation ergonomique des interfaces homme-machine : une revue de la littérature

Bernard Senach

INRIA

UNITÉ DE RECHERCHE
INRIA-SOPHIA ANTIPOLIS

Institut National
de Recherche
en Informatique
et en Automatique

Domaine de Voluceau
Rocquencourt
B.P.105
78153 Le Chesnay Cedex
France
Tél (1) 39 63 55 11

# Rapports de Recherche

*1992*

*ème anniversaire*

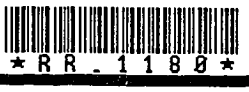
**N° 1180**

***Programme 8***
*Communication Homme-Machine*

# EVALUATION ERGONOMIQUE DES INTERFACES HOMME-MACHINE : UNE REVUE DE LA LITTERATURE

**Bernard SENACH**

**Mars 1990**



# Evaluation ergonomique des Interfaces Homme-Machine:

# une Revue de la Littérature

# Ergonomics and Evaluation of Man-Computer Interfaces: a Review

*Bernard Senach*

*I.N.R.I.A.*
*Centre de Sophia Antipolis*
*Route des Lucioles*
*06560 Valbonne - FRANCE*

*7 Février 1990*

## Résumé

Les ergonomes réalisent en général des évaluations d'interfaces homme-machine (IHM) dans 4 contextes principaux:

- pour établir un *diagnostic d'usage* de systèmes existant
- pour *assurer la qualité* de la conception d'une IHM
- pour *comparer les avantages et les inconvénients* de logiciels verticaux
- pour *contrôler a priori la qualité* ergonomique d'un produit commercialisé

Chacun de ces contextes pose des questions originales et présente des contraintes spécifiques nécessitant la mise en jeu de techniques bien adaptées. Ce texte propose une revue critique des pratiques actuelles de l'évaluation d'interface. On discute tout d'abord les *approches empiriques* basées sur la mesure des performances de l'utilisateur en rapportant quelques travaux et en montrant leurs limites. Les *approches analytiques* basées sur des modèles de l'interface et de l'interaction homme-machine sont ensuite considérées.

L'analyse fait apparaître les principales directions de travail actuellement engagées et les dimensions d'évaluation qui intéressent l'ergonome.



## Abstract

The evaluation of Man-Computer Interfaces (MCI) is generally driven by four main classes of objectives:

- the *diagnosis of the usability* of an existing MCI
- the *improvement of the MCI during design*
- the *comparison of the strong and weak points* of several software in one problem domain
- the *a priori evaluation of the ergonomic quality* of an MCI.

Each of theses objectives requiers well-designed evaluation technics. In this paper we propose a critical review of the present know-how in MCI evaluation. We discuss first the *empirical approach* based on measurement of man-machine performances through some recent reasearch and we show their limits. Then, we consider the *analytical approach* based on formal models of the MCI and of the interaction.

## PRÉSENTATION GÉNÉRALE



## CHAPITRE I.

## APPROCHE EMPIRIQUE DE L'ÉVALUATION

## CHAPITRE II.

## EVALUATIONS COMPARATIVES DE LOGICIELS VERTICAUX



## CHAPITRE III.

## APPROCHES ANALYTIQUES DE L'ÉVALUATION
EVALUATION A PRIORI DE LA QUALITÉ D'UNE IHM

## CONCLUSION

# EVALUATION ERGONOMIQUE DES INTERFACES HOMME-MACHINE

## UNE REVUE DE LA LITTÉRATURE

# 1. Introduction

Les outils de développement proposés dans les ateliers de génie logiciel actuels améliorent sensiblement l'homogénéité des Interfaces Homme-Machine (IHM):

- les entités interactives des boîtes à outils contribuent jusqu'à un certain point à la cohérence intra et inter-application. L'utilisateur retrouve au niveau des logiciels développés dans le même environnement des objets dont les attributs graphiques et le comportement de surface sont standardisés

- les guides de style associés aux outils de développement proposent en même temps au concepteur un ensemble de recommandations nettement plus faciles à exploiter que les habituelles compilations de règles ergonomiques.

Ces outils ne garantissent cependant pas encore à l'utilisateur final la facilité d'apprentissage et d'utilisation: il reste en effet fréquent que les interfaces utilisateur de systèmes réputés "conviviaux" tels que le Macintosh posent les mêmes problèmes de communication que des applications classiques organisées en "page écran". Par exemple: les informations requises pour une tâche sont présentées dans des contextes de travail exclusifs (travail "en aveugle"), des effets de bord non contrôlés déterminent la perte de données, ou encore les affichages sont ambigus ou incompréhensibles (messages d'erreur, aide en ligne...). La persistance de ces difficultés n'est pas due à la sous-estimation des contraintes ergonomiques qui sont maintenant reconnues comme étant essentielles à l'amélioration des performances des systèmes homme-machine; le problème tient plutôt à ce que les travaux d'ergonomie du logiciel sont encore souvent trop qualitatifs pour être exploitables.

De plus, les méthodes usuelles pour "prendre en compte l'utilisateur" dès la conception sont lourdes à mettre en oeuvre au regard des exigences de rapidité de développement: elles consistent à compléter les analyses informatiques par des analyses du travail plus ou moins élaborées pouvant aller jusqu'à la modélisation de l'activité cognitive de l'utilisateur. Si les ergonomes désirent voir appliquées les idées qu'ils défendent, ils doivent déterminer de nouvelles méthodes permettant d'intégrer les contraintes ergonomiques dès les premières phases de la conception. Une approche pragmatique est d'enrichir les environnements de développement d'IHM par des services ergonomiques tels que l'évaluation automatique de leur qualité pour l'utilisateur, la proposition d'alternatives de conception, et/ou l'explication de critères de choix entre plusieurs solutions...

Une solution technique évitant les "bugs d'interface" est d'ores et déjà disponible dans certains environnements de conception comportant des systèmes experts en ergonomie (voir par exemple Waldhör, 1989). L'analyse présentée ci-dessous a été réalisée dans le cadre d'un projet visant à terme à mettre à la disposition d'opérateurs n'ayant pas une grande expérience de l'ergonomie de l'informatique un outil d'évaluation des IHM[1]. Une première étape a consisté en un recensement des techniques actuellement utilisées pour évaluer la qualité ergonomique d'un logiciel. C'est cette analyse qui est présentée ci-dessous.

Le rapport comporte trois chapitres abordant successivement les approches empiriques, les évaluations comparatives et les modèles analytiques.

---

[1] L'étude bibliographique a été réalisée pour la conception du système expert ERGOVAL (contrat établi entre le Service de Recherche Technique de la Poste et la Société ILOG , l'auteur étant Conseiller Scientifique d'ILOG).

# 2. A propos de l'évaluation

Toute évaluation consiste à comparer un *modèle de l'objet évalué* à un *modèle de référence* permettant d'établir des conclusions. Plus précisément, une évaluation est réalisée dans le cadre d'un *objectif* qui détermine les dimensions d'analyse pertinentes. Celles-ci constituent le modèle de l'objet évalué et nécessitent la définition de *variables cibles* et de *techniques de recueil* de ces variables. C'est ce que résume le schéma ci-dessous:

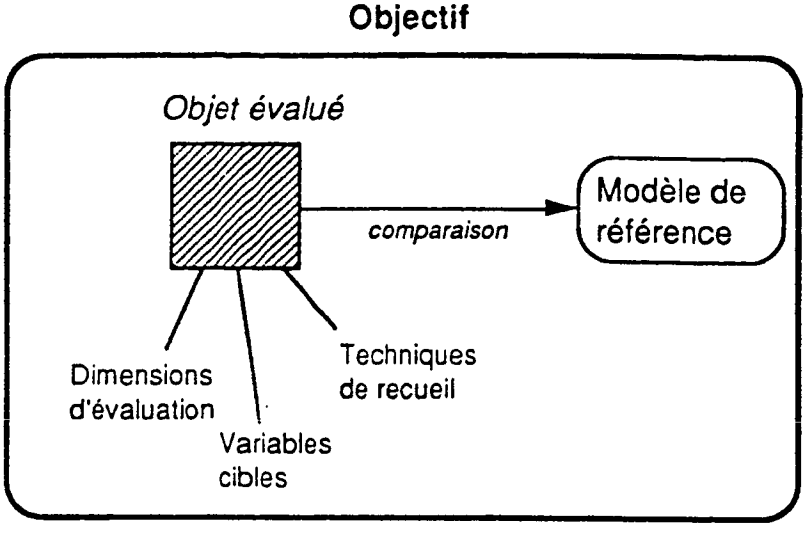



*Fig. 1: Schéma de principe d'une évaluation*

Les pratiques en matière d'évaluation sont entièrement définies par la spécification de chacun de ces points. Avant d'aborder le détail des différentes approches actuelles, on distingue quelques dimensions qui organisent l'analyse.

## 2.1. Objectifs d'évaluation

Les objectifs d'évaluation sont exprimés sous la forme de questions auxquelles l'ergonome devra être en mesure de répondre à l'issue de l'analyse. Ces questions peuvent être situées à des niveaux d'abstraction très différents et concernent des champs plus ou moins étendus de l'interaction homme-machine. En voici quelques-unes des plus usuelles:

- est-ce qu'un dispositif donné est utilisé ou non ?
- pourquoi un dispositif donné est-il sous-utilisé ?
- quelles sont les difficultés que rencontrent les utilisateurs ?
- les menus sont-ils bien organisés ?
- quel logiciel faut-il préférer pour réaliser au mieux une tâche donnée ?
- comment peut-on améliorer la présentation d'information sur un écran ?
- l'interface permet-elle à l'utilisateur de réaliser sa tâche correctement ?

Certaines de ces interrogations sont très spécifiques et concernent une interface déterminée, d'autres sont moins bien définies et ont une portée plus générale, voire universelle dans la mesure où elles concernent potentiellement tous les systèmes homme-machine. Lorsqu'un énoncé de problème est formulé à un niveau de généralité trop élevé, l'évaluation est dirigée par des objectifs mal définis et la plupart du temps les résultats sont difficilement exploitables: on découvre *a posteriori* que les données recueillies ne permettent pas de répondre aux questions que l'on voulait traiter. Le niveau de détail de l'objectif est ainsi un déterminant de la qualité d'une évaluation. Le rapport s'intéresse à 4 classes principales d'objectifs:

- l'analyse de l'utilisation d'un dispositif en situation de travail
- la sélection d'alternatives de conception
- la détection et la correction des défauts d'une IHM
- l'évaluation comparative de différents logiciels

## 2.2. Modèle de l'objet évalué

### *2.2.1. Deux dimensions d'une évaluation*

Il est parfois étonnant pour un ergonome que des interfaces ayant de médiocres qualités ergonomiques soient très appréciées de leurs utilisateurs[1]. Par exemple c'est typiquement le cas de "Visicalc", le premier tableur commercialisé qui, malgré des performances techniques sous-optimales (en particulier au niveau du temps de réponse[2]) a rencontré un succès commercial considérable. Une des raisons de cet apparent paradoxe tient à ce que malgré leurs faibles qualités ergonomiques ces dispositifs apportent des services essentiels.

A l'inverse, il arrive que des logiciels soient peu utilisés bien qu'ils proposent des interfaces développées dans des environnements évolués (boîtes à outils, UIMS...) offrant toutes les facilités actuelles de manipulation directe, de contrôle direct de l'effet des commandes (Wysiwyg), d'adaptabilité de l'interface et respectant des contraintes ergonomiques de surface (cohérence syntaxique et lexicale, codes non ambigus, présentation d'information structurée...). Cette sous-utilisation tient parfois à ce que des fonctionnalités jugées importantes n'ont pas été implémentées ou que les tâches sont organisées d'une façon tellement différente des pratiques habituelles que le gain estimé au regard du coût d'apprentissage est jugé insuffisant.

Enfin, les logiciels ayant des capacités fonctionnelles impressionnantes (UNIX, MACSYMA....) se révèlent souvent très difficiles à utiliser et l'apprentissage nécessaire pour en acquérir la maîtrise est parfois décourageant. Cette complexité fonctionnelle détermine en général une sous-utilisation dont on trouvera dans Eason (1984) une illustration quantifiée dans un contexte professionnel: une analyse de l'utilisation d'un logiciel de gestion bancaire montre que sur les 36 fonctions proposées aux opérateurs pour réaliser leurs tâches, 4 d'entre elles représentent 75 % de l'utilisation totale.

Ce triple constat montre tout d'abord que toute analyse devrait considérer la *motivation* de l'utilisateur: cette notion constitue en psychologie un facteur déterminant de l'apprentissage et de toute évidence, un utilisateur bien motivé peut supporter bien des défauts d'une interface. Pour éclaircir les relations complexes entre la richesse des fonctionnalités d'un logiciel, leur intérêt pour l'utilisateur et la facilité d'utilisation, il convient de différencier deux dimensions principales d'une évaluation: celle qui concerne l'*utilité* du produit et celle qui s'intéresse à *qualité de son interface*, i.e. son "utilisabilité" selon la formulation de Shackel (1984).

### *2.2.2. Utilité et "utilisabilité"*

La première dimension détermine si le produit permet à l'utilisateur d'atteindre ses objectifs de travail. Elle porte sur des propriétés telles que la capacité fonctionnelle, les performances du système et la qualité de l'assistance technique proposée au client. Elle n'est pas traitée en détail mais est abordée à propos de l'évaluation comparative. La seconde dimension concerne la qualité de l'interaction homme-machine, i.e. la facilité d'apprentissage et d'utilisation.

Le barbarisme "d'utilisabilité" traduit le concept anglais "usability" développé par Eason (1984) pour rendre compte du paradoxe suivant: au lieu de faciliter l'usage des logiciels, la multiplication des fonctionnalités offertes à l'utilisateur a pour effet de lui rendre la tâche de plus en plus compliquée. L'"utilisabilité" sert ainsi à poser la frontière entre utilité potentielle et utilité réelle. Dans la mesure où

---

1 Voir à ce sujet la polémique développée autour de l'interface d'Unix (Norman, 1981, Compton, 1984).

2 Développé par des informaticiens amateurs mais professionnels du domaine d'application, la vitesse de traitement a pu être améliorée de 30% par des développeurs expérimentés. La qualité du modèle conceptuel sous-jacent peut ainsi pondérer l'importance des médiocres performances techniques.

l'adjonction de fonctions censées faciliter l'exploitation d'un système multi-fonctionnel a pour effet pervers de le rendre un peu moins utilisable, il devient de plus en plus critique de déterminer à quelles conditions les utilisateurs peuvent se servir des possibilités mises à leur disposition. On adopte ici un point de vue "systémique" selon lequel l'utilisabilité d'un système constitue une réponse de l'utilisateur à ses propriétés générales: elle ne dépend pas de caractéristiques locales ("la structure de commande 1 est meilleure que la 2", ou bien "le dispositif d'entrée 1 est préférable au dispositif d'entrée 2") mais de la "cohérence de la conception":

- la "cohérence interne" d'une IHM est établie par la régularité des décisions prises par le concepteur et se traduit par des aspects aussi divers qu'un langage de commande bien structuré ou des positions invariantes de l'affichage des mêmes informations d'un écran à l'autre.... Cette cohérence détermine la qualité intrinsèque de l'interface homme-machine (IHM).

- la "cohérence externe" est établie par rapport à des critères d'adéquation relatifs aux tâches et au fonctionnement cognitif des utilisateurs (conditions d'apprentissage, stratégies de raisonnement..)

Le schéma ci-dessous résume ce point de vue général:

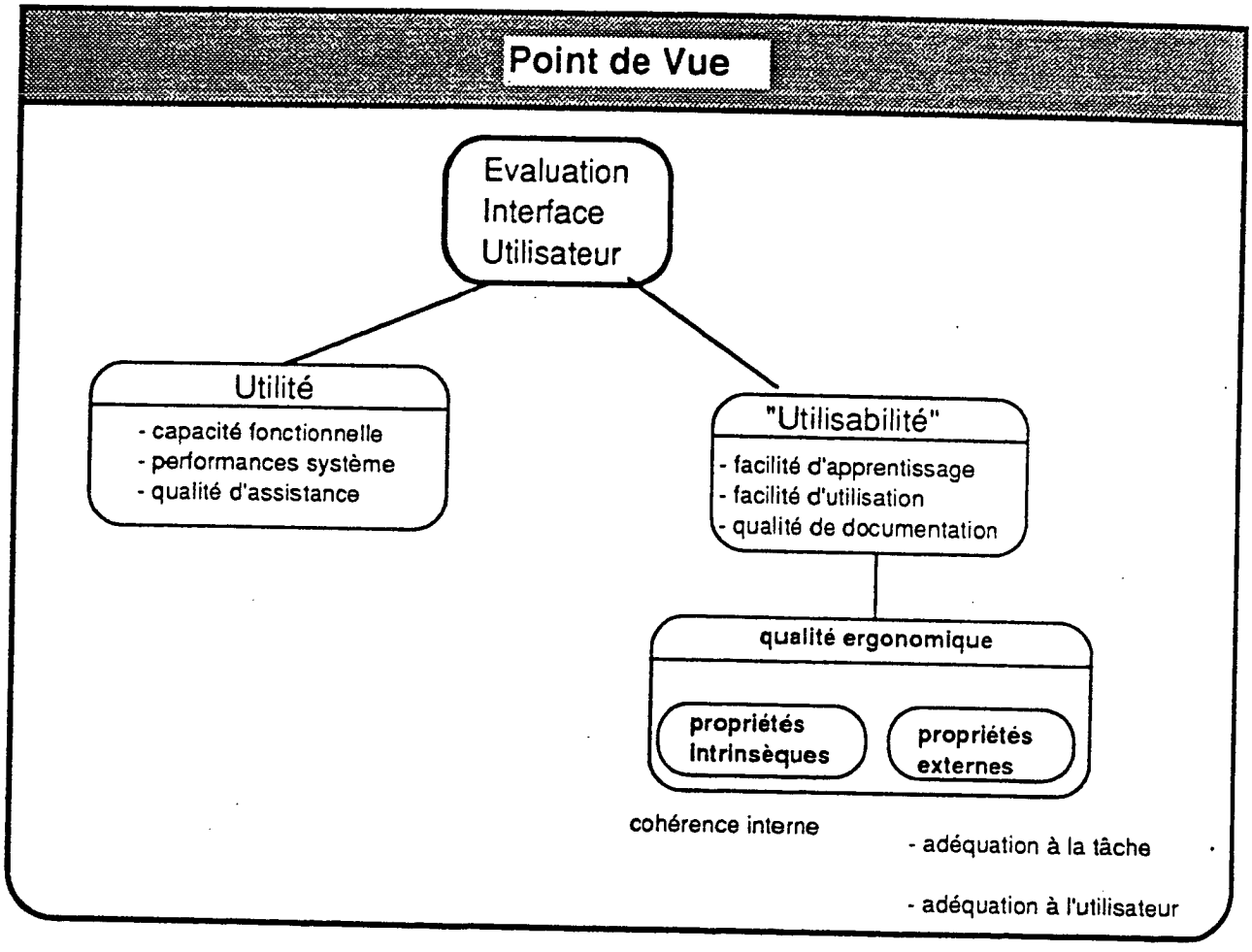



*Fig. 2: Dimensions de l'évaluation d'une IHM*

## 2.3. Variables cibles

L'abstraction des propriétés d'un objet sur lesquelles va porter l'analyse définit un modèle de l'objet évalué. Les attributs et les propriétés sur lesquels vont être réalisées les mesures sont partiellement

contraints par les objectifs poursuivis. Par exemple, le poids, la forme d'un objet... peuvent être des attributs pertinents si l'on veut établir sa facilité de transport et la couleur ne sera prise en considération que si l'on s'intéresse à son esthétique. Le choix d'un modèle d'IHM lors d'une évaluation est dirigé par le contexte; autrement dit, la sélection des dimensions pertinentes pour un diagnostic de qualité ergonomique dépend non seulement des objectifs de l'évaluation mais aussi des caractéristiques de la population et des exigences des tâches. Par exemple, la qualité de la documentation peut être essentielle dans certains environnements de travail alors que le temps de réponse du système devient l'un des facteurs critiques dans d'autres situations.

Lorsque les dimensions d'évaluation sont établies, il convient de définir les variables qui vont être enregistrées. Les variables psychologiques mesurant la facilité d'apprentissage et d'utilisation ne peuvent pas être appréhendées directement et doivent être opérationalisées. La taxonomie ci-dessous présente quelques unes des principales variables dépendantes utilisées:

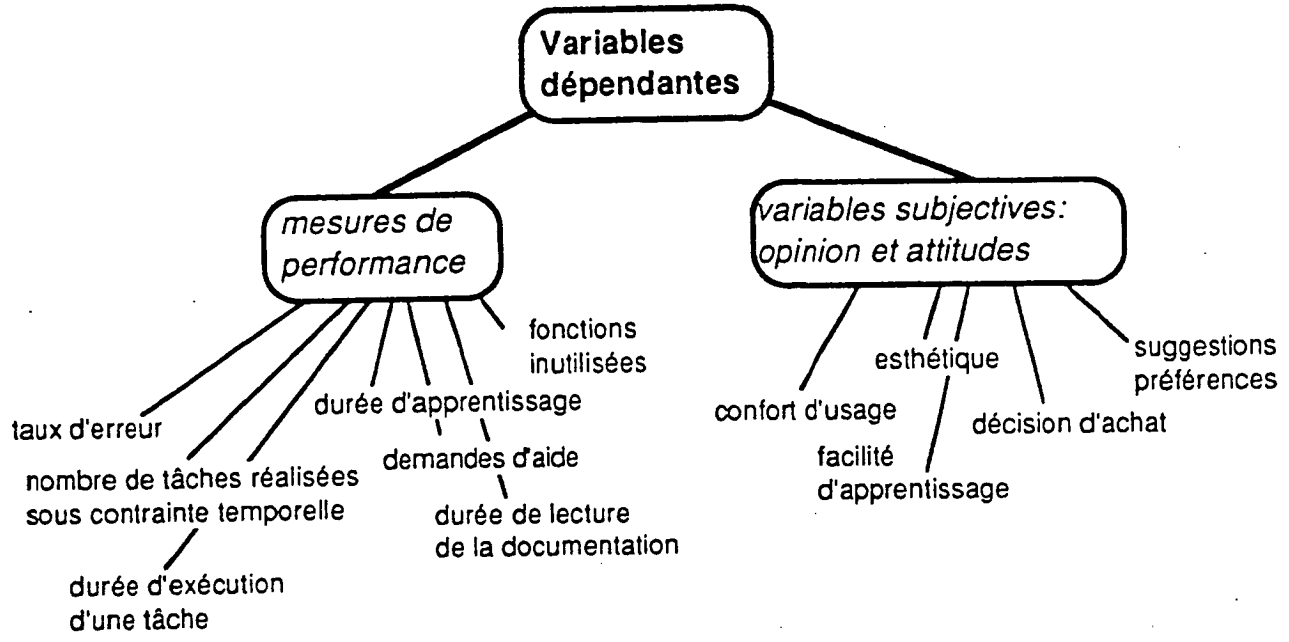



*Fig. 3: Principales variables utilisées lors de l'évaluation d'IHM*

La construction d'un outil de mesure pose des problèmes bien connus de validité (est-ce l'outil mesure avec exactitude la caractéristique en question), de fidélité (variabilité des résultats en fonction des évaluateurs) et de précision (sensibilité des mesures aux variations observées). Le point le plus important pour l'évaluation des interfaces concerne la *validité écologique* des mesures. On donnera ci-dessous de nombreux exemples de cette question récurrente.

## 2.4. Techniques de recueil des variables cibles

Toutes les évaluations d'une IHM visent à identifier ou à prévoir les difficultés que rencontrent les utilisateurs et à en caractériser les points forts et les points faibles. A ce titre, on pourrait supposer que les techniques d'évaluation disponibles sont toutes également valides quels que soient les contextes dans lesquels se déroulent les études.

L'analyse de la littérature montre que les évaluations entreprises ne sont pas toujours couronnées de succès. Les raisons peuvent tenir tout d'abord à ce que les objectifs poursuivis ne sont pas clairement définis, mais il existe aussi une forte dépendance entre les questions initiales et les méthodes (cliniques et/ou expérimentales) mises en oeuvre. L'évaluation peut être réalisée à partir de techniques très diverses: questionnaires, monitoring, protocoles verbaux, enregistrement vidéo.. Les échecs des évaluations tiennent souvent à ce que les techniques utilisées ne sont pas adaptées au projet de l'évaluateur (Sweeney et Dillon 1987). A titre d'illustration, les techniques cliniques utilisées dans les études diagnostic (par exemple les protocoles verbaux) sont souvent d'un faible secours lors des tests itératifs réalisés en cours de conception (voir par exemple, Lund 1985) car les questions qui se posent

dans ce contexte sont souvent très spécifiques: il s'agit par exemple de choisir des alternatives (quelle est la meilleure organisation des menus ?) ou de valider des choix incertains (la sélection par double-clic pose-t-elle des problèmes particuliers ?) et les approches expérimentales, mieux structurées sont plus adaptées. D'autre part, le "diagnostic" des points faibles d'un dispositif établi pour développer une version améliorée n'est le même que celui qui est posé par le banc d'essai d'un produit fini avant sa commercialisation.

Pour mieux cerner cette relation entre les objectifs poursuivis et les techniques mises en oeuvre, on propose ci-dessous une classification des principaux contextes d'évaluation auxquels est régulièrement confronté l'ergonome. Chacun de ces contextes pose des questions originales et présente des contraintes spécifiques nécessitant la mise en jeu de techniques bien adaptées. On a distingué ici:

- le diagnostic d'usage de système existant
- les tests réalisés en cours de conception
- les évaluations comparatives de logiciels verticaux
- le contrôle *a priori* de la qualité de l'interface

## 2.5. Modèle de référence: approche empirique et approche analytique

Le modèle de référence auquel sont comparées les données enregistrées fonde les conclusions de l'évaluation: ayant déterminé la valeur des attributs, que peut-on en dire du point de vue de la question initiale ?

Les deux premières situations d'évaluation mentionnées au paragraphe précédent (diagnostic d'usage et les tests de conception) sont différenciés par le fait que dans le premier contexte, il existe une *expérience de l'utilisation* du logiciel évalué qui par définition n'existe pas dans un contexte de conception. Elles constituent un contrôle *a posteriori* et reposent sur une approche empirique i.e. sur le recueil et l'analyse de **données comportementales**. Dans cette perspective, il ne s'agit pas de caractériser la qualité technique d'un produit manufacturé (portabilité, efficacité, fiabilité...) mais d'enregistrer des données rendant compte de son utilisation. Ces données permettent ensuite d'inférer (avec plus ou moins de bonheur selon la précision et la validité des mesures effectuées) les difficultés que rencontreront les utilisateurs et de développer des solutions les réduisant. Cette approche met en jeu les multiples outils élaborés par la méthode expérimentale, c'est à dire pour l'essentiel: test d'hypothèses, recueil contrôlé des données, traitements statistiques et interprétation des résultats au regard de critères d'évaluation bien spécifiés.

Le contrôle de la qualité d'une IHM est défini ici comme une évaluation *a priori* des caractéristiques ergonomiques de l'interface. Il nécessite une approche analytique i.e. une évaluation faite sur un **modèle de l'interface** ou de l'interaction. Le recours à des représentations abstraites autorise des prédictions relatives aux performances qui ne peuvent pas être établies dans une approche purement empirique.

Les évaluations comparatives concernent les logiciels permettant de réaliser la même classe de tâches (éditeurs de texte, comptabilité, gestion...). Elles nécessitent la mise au point de bancs d'essais de tâches exploitant des modèles analytiques. Le découpage entre approche empirique et approche analytique est réalisé pour la commodité de l'exposé car en réalité, les points de vue ne sont pas opposés mais complémentaires. Par exemple les études analytiques exploitent les résultats d'études empiriques pour valider leurs prédictions et les approches empiriques conduisent à élaborer des hypothèses qui contribuent au développement des modèles analytiques.

## 2.7. Résumé

Ces distinctions sont résumées dans le schéma ci-dessous:

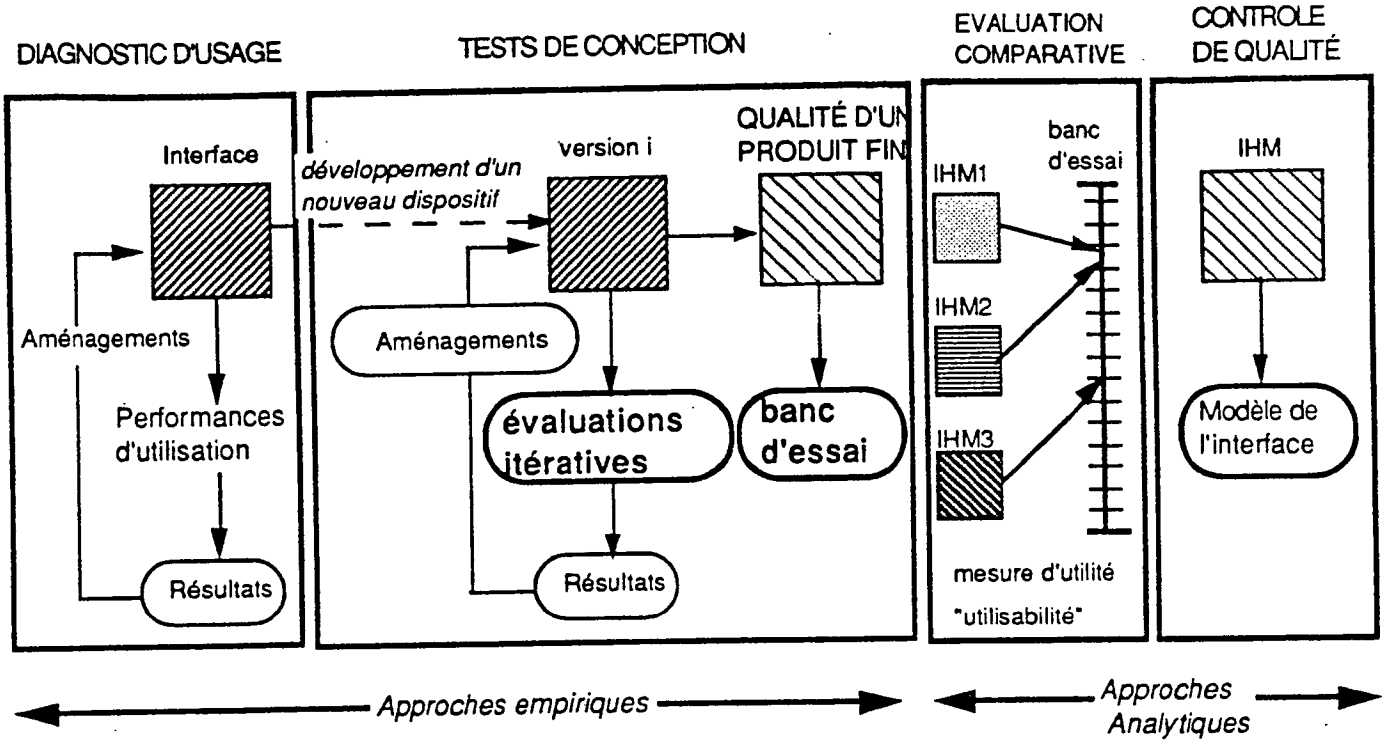



*Fig. 4: 4 contextes d'évaluation des interfaces utilisateur*

Le chapitre suivant présente les approches empiriques en discutant les principales techniques utilisées, puis on analyse les évaluations comparatives à travers deux illustrations, la première concernant les éditeurs de texte, la seconde proposant une méthodologie plus générale. Les modèles de l'interface et de l'interaction supportant les approches analytiques sont ensuite abordés.

# Chapitre I.

# Approche empirique de l'évaluation

La structure du chapitre traitant des approches empiriques est décrite dans l'arbre ci-dessous:

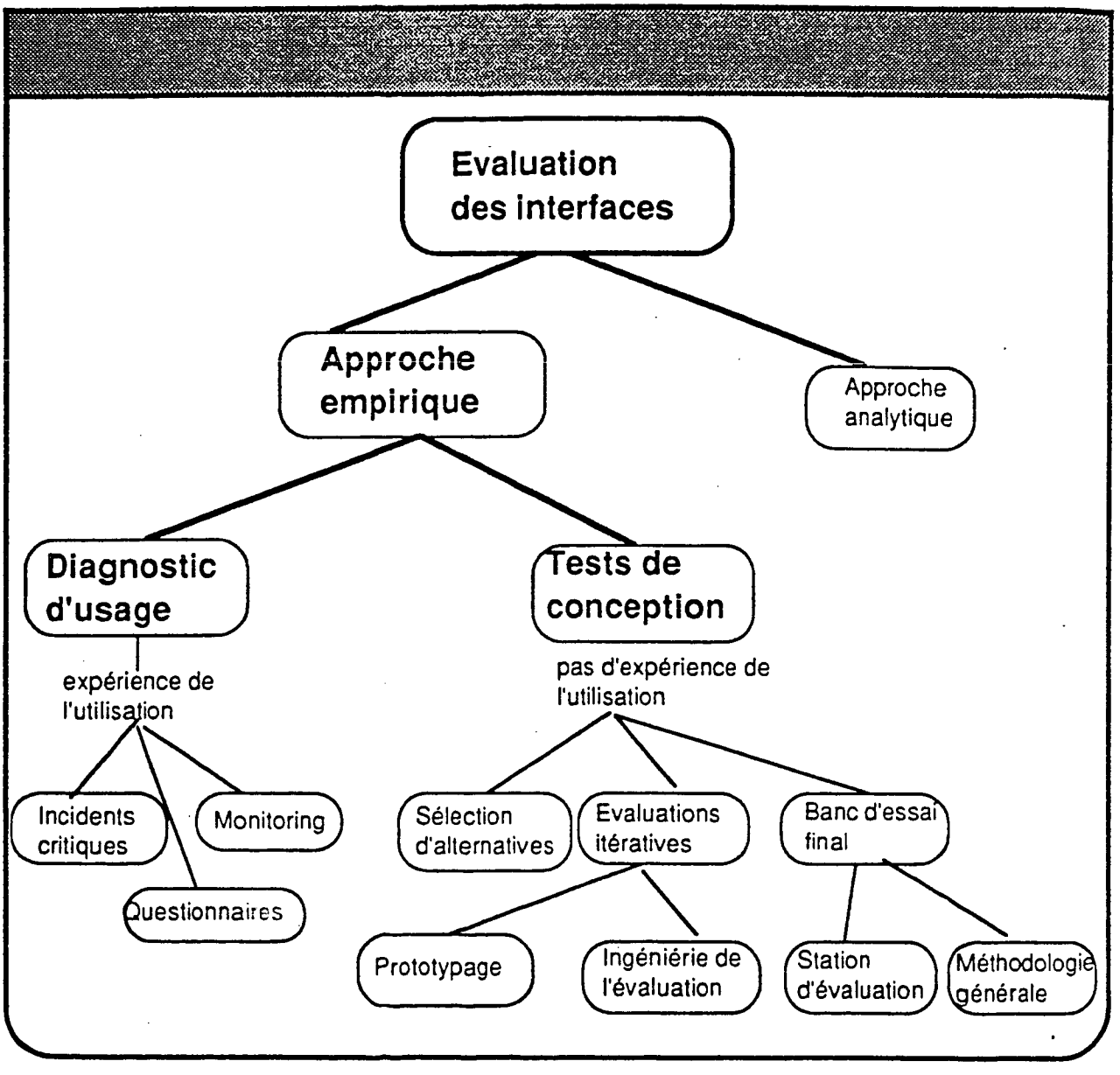



*Fig. 5: Structure du Chapitre I*

# I. Diagnostic d'usage d'un système existant

Lorsque l'évaluation concerne un dispositif existant, celui-ci a généralement fait l'objet d'une utilisation intensive. Si les utilisateurs peuvent être impliqués dans les tests, l'analyse doit s'appuyer sur cette *expérience d'usage*: les données recueillies traduisent des pratiques bien structurées et permettent alors d'identifier les modalités d'exploitation réelles du dispositif en situation de travail. Le choix d'une technique de diagnostic d'utilisation dépend beaucoup de contraintes circonstancielles (durée d'intervention, matériel disponible...). Les trois techniques présentée ci-dessous (incidents critiques, questionnaires d'utilisation et mouchards électroniques) correspondent à des différentes phases d'une intervention ergonomique dans lesquelles les informations requises pour l'analyse du système homme-machine sont définies à des niveaux de détail de plus en plus important.

## 1. Approche clinique du diagnostic d'usage: les Incidents critiques

Les techniques cliniques d'observation et d'entretien sont bien adaptées aux *approches exploratoires* et sont nécessaires lors de "l'analyse diagnostique" de systèmes complexes. Typiquement, dans le domaine du contrôle de processus, l'environnement de travail est technologiquement saturé et les opérateurs doivent intervenir sur des dispositifs techniques très hétérogènes (différents constructeurs, différents modèles, plusieurs sous-processus...). Dans la première phase d'analyse de tels contextes de travail, l'évaluation de la qualité ergonomique d'un dispositif particulier n'a pas grand sens et des techniques cliniques sont très utiles pour détecter les problèmes et identifier ceux qui devront faire l'objet d'analyses ultérieures. Ces techniques sont en général difficiles à mettre en oeuvre car elles reposent sur des heuristiques; l'analyse des incidents critiques étant structurée, elle évite quelques unes des difficultés usuelles.

### *1.1. Principe*

La technique consiste en un *recueil systématique des dysfonctionnements d'un système homme-machine* à partir d'entretiens avec les utilisateurs et d'observations conduites en situation naturelle (ie sur le poste de travail des opérateurs). Ces incidents sont décrits sous la forme de courts récits décrivant les faits. Ils font ensuite l'objet d'une classification hiérarchique ascendante dans laquelle on regroupe dans un premier temps des incidents qui sont des instances d'un même problème (par exemple confusion de commandes dans le contexte C1 puis dans le contexte C2). Les problèmes sont ensuite regroupés en des classes plus générales (par exemple, la classe "ambiguïtés du système de contrôle-commande" comporte les problèmes "confusions de commandes", "interprétation erronée d'un affichage", "absence de retour d'information"...). L'identification de ces classes permet de rechercher des solutions ayant une portée globale. Par exemple, les informations acquises à partir des incidents critiques peuvent être utilisées pour:

- définir les premières fonctionnalités d'un nouveau système
- réaliser des aménagements (logiciels, matériels, documentaires)
- préciser les objectifs de formation...

### *1.2. Avantages et limites*

Ces analyses cliniques permettent d'obtenir rapidement un *diagnostic global* des dysfonctionnements d'un environnement de travail donné. Elles fournissent le contexte d'interprétation correct des difficultés d'utilisation d'un dispositif: celles-ci ne sont souvent qu'un *symptôme* de l'inadéquation des outils à la logique du travail des opérateurs. Le lecteur intéressé pourra consulter deux études techniques conduites en secteur industriel (Senach et Alengry, 1985; Senach et Pichancourt, 1986) illustrant des évaluations prenant en compte de l'environnement socio-technique. En l'occurrence, dans les contextes étudiés, les utilisateurs sont amenés à travailler avec divers dispositifs de contrôle-commande (machine-outil, base données, systèmes d'assistance...), l'évaluation doit alors s'appuyer d'une part sur l'analyse des interactions avec les autres outils (du fait notamment des problèmes de transfert de procédure) et d'autre part sur

l'identification des exigences de l'activité des opérateurs. Par exemple, la nécessité de déplacements fréquents définit quelques critères d'évaluation du dispositif: celui-ci doit par exemple comporter des facilités de gestion des interruptions (gel des affichages, historique des commandes précédentes...) et de consultation à distance (boîtier de contrôle-commande décentralisé...). Cette *perspective écologique* évite d'aménager localement un dispositif lorsque ce sont les contraintes du système socio-technique qui empêchent son utilisation efficace.

Les approches cliniques facilitent le "débroussaillage" des systèmes homme-machine complexes et conduisent à un diagnostic global, mais les techniques utilisées restent *trop macroscopiques* pour évaluer précisément la qualité ergonomique d'un dispositif particulier. Lorsque la focale de l'analyse doivent être modifiée, d'autres techniques sont mieux adaptées: le *questionnaire d'utilisation* permet d'obtenir des informations plus précises sous certaines conditions.

# 2. Questionnaires et diagnostic d'usage

Le questionnement de l'utilisateur est une pratique commune à la plupart des situations d'évaluation mais il est envisagé selon des perspectives très diverses. L'administration de questionnaires pré et post-expérimentaux permet d'identifier les *modifications d'attitude* après l'utilisation d'un dispositif (Sweeney et Dillon, 1987), les *enquêtes de satisfaction* sont conduites d'un point de vue prospectif [1] ou visent directement au contrôle de qualité de l'IHM (Smith et Mosier, 1984). Ces derniers suggèrent une procédure vérifiant la satisfaction des besoins des utilisateurs dans laquelle il s'agit dans un premier temps de leur faire établir une liste d'exigences, puis d'en dériver par des techniques statistiques l'"opinion commune". L'analyse consiste dans une seconde phase à vérifier systématiquement l'accord entre les demandes enregistrées et les caractéristiques du produit. Ce type d'évaluation est abordé plus loin dans le cadre de l'approche analytique (cf grilles d'évaluation ergonomique p. 39).

On s'intéresse ici aux *questionnaires d'utilisation* qui complètent les études expérimentales par le recueil d'appréciations subjectives concernant le dispositif et qui contribuent au contrôle et à la validation des résultats expérimentaux obtenus par ailleurs. On en trouvera des exemples dans Kellog (1987) qui utilise une batterie de questionnaires pour identifier les effets de la cohérence conceptuelle sur la facilité d'usage (cf. p. 54), ou encore dans Tullis (1984) qui exploite les opinions sur la qualité d'écrans pour valider un modèle théorique de la complexité perceptive (cf. p.58).

## *2.1. Avantages et inconvénients des questionnaires*

Le point discuté ici est de déterminer dans quelle mesure un questionnaire permet d'établir un diagnostic d'usage, i.e. si des informations relatives à l'utilisation d'un dispositif acquises a posteriori sont réellement exploitables. On peut se demander en effet si l'utilisation d'un *matériel statique* (papier-crayon) pour analyser une situation essentiellement dynamique ne biaise pas le recueil car il peut être difficile de se rappeler précisément les difficultés rencontrées lors de l'utilisation (Root et Draper, 1983). Par ailleurs, si l'outil présente l'avantage évident de permettre le *recueil économique* d'un ensemble important d'informations, il n'est pas toujours d'un emploi évident. Plusieurs types de questionnaires peuvent être élaborés (questionnaires "opératifs" à propos de la sémantique et de la syntaxe des commandes, différenciateur sémantique, évaluation subjective sur un ensemble d'adjectifs pré-définis...) et la première difficulté est de choisir le plus adapté au problème à traiter. La construction du questionnaire n'est pas non plus triviale: elle dépend uniquement de l'expérimentateur, ce qui peut déterminer de nombreuses difficultés d'analyses ultérieures (voir à ce sujet Grawitz, 1974): une sélection trop restrictive peut laisser de côté des questions importantes, ou bien une formulation trop floue peut conduire à ne pas très bien savoir ce que l'on mesure exactement [2]. Enfin, il y a peu de questionnaires dédiés à l'évaluation

---

[1] Voir par exemple à ce sujet Rushnick & al. (1986) qui cherchent à hiérarchiser les critères de satisfaction des utilisateurs et Perlman (1985a) qui propose un formalisme pour structurer les enquêtes électroniques.

[2] A titre d'illustration, une des conclusions d'une enquête récente conduite auprès d'utilisateurs énonce la règle "le bouton de la souris doit toujours être sous le doigt de l'utilisateur".

d'interface qui aient été validés par des contrôles rigoureux et les conclusions qu'on en tire devraient être établies avec prudence du fait de l'absence de standardisation.

L'intérêt des questionnaires comme outil de diagnostic d'usage a été étudiée expérimentalement par Root et Draper (1983).

### *2.2 .Etude expérimentale de l'intérêt des questionnaires comme outils d'évaluation des interfaces*

Les auteurs ont étudié l'importance de trois facteurs sur la qualité des réponses des utilisateurs:

- l'effet des différents types de question (questions ouvertes ou spécifiques)
- l'effet des connaissances préalables des sujets (transfert de connaissances)
- l'effet de la méthode d'administration (papier-crayon, questionnaire en ligne)

L'objectif de l'analyse est de contrôler la connaissance d'un éditeur et de mettre en évidence les faiblesses de son interface. L'expérience consistait à faire évaluer par des étudiants l'éditeur de programme à partir duquel ils apprennent un langage évolué (Pascal UCSD). Deux conditions expérimentales ont été testées: l'une dans laquelle les sujets remplissaient un questionnaire papier-crayon, l'autre dans laquelle une partie du questionnaire était implémentée sur ordinateur. Divers groupes de contrôle subissaient le test. On retiendra ici de ce travail la structure du questionnaire utilisé. Il regroupe trois types de questions:

- une grille d'évaluation des commandes
- des questions spécifiques
- des questions ouvertes

a) Grille d'évaluation des commandes

La grille concerne l'ensemble des commandes de l'éditeur. Pour chacune d'entre elles, le sujet doit indiquer sur une échelle en trois points si:

- elle est connue
- elle est utilisée ou évitée
- elle est dangereuse
- son utilisation est malcommode
- elle est difficile à taper
- elle a une syntaxe difficile
- il est difficile d'en prédire les résultats

b) Questions spécifiques

Les sujets doivent évaluer l'intérêt que présenteraient certaines modifications de l'éditeur en notant leur utilité. Par exemple: "indiquer l'importance que représente pour vous ce changement") et exprimer des opinions a propos de caractéristiques particulières de l'éditeur (commandes ou autre) ("aimeriez vous avoir une commande d'effacement de mots ?").

c) Questions ouvertes

Elles demandent au sujet d'exprimer les difficultés rencontrées et d'indiquer les modifications qu'ils souhaitent dans une version améliorée de l'éditeur. Certaines questions visent à établir la connaissance de l'éditeur: pour ces questions, le sujet doit définir la séquence de commandes réalisant une tâche d'édition déterminée et prédire l'effet d'une séquence donnée sur un texte.

Le schéma général que l'on peut en donner est le suivant:

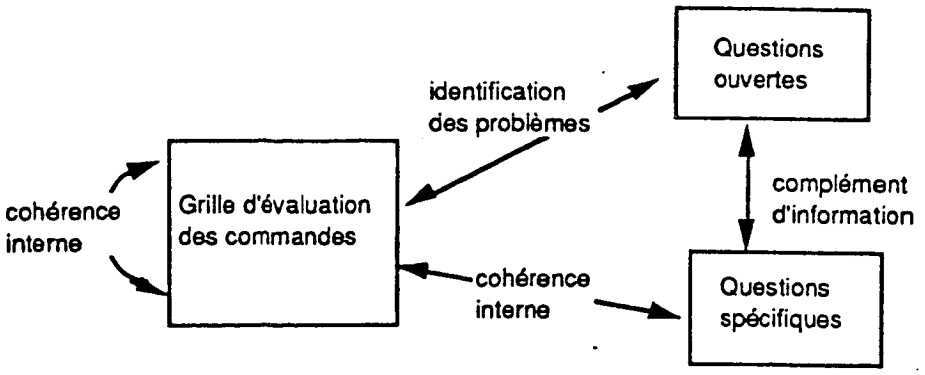



*Fig. 6: Organisation du questionnaire utilisé par Root et Draper (1983)*

Le point le plus intéressant est que la structure du questionnaire autorise la *vérification de la cohérence* des réponses; plusieurs techniques sont utilisées: formulations différentes référant aux mêmes items, références croisées entre les questions de types différents, dispersion des questions relatives à une même commande et nature "opérative" de certaines questions (prédiction des effets, description de la séquence d'action pour la mise en oeuvre...). Ces procédés améliorent considérablement la qualité des réponses fournies.

d) Principaux résultats

Sans entrer dans les détails de l'analyse, on retiendra ici que, sous réserve qu'il fasse l'objet d'une construction minutieuse et qu'il soit soumis à validation, le questionnaire d'utilisation constitue un outil d'évaluation intéressant:

- la grille d'évaluation des commandes comporte les questions les plus utiles: elles identifient la connaissance des commandes en la traduisant par un pourcentage mettant en évidence les commandes inconnues des utilisateurs. Ces questions ne peuvent cependant être appliquées qu'à des caractéristiques existantes: la qualité des réponses recueillies dépend avant tout de l'expérience qu'en ont les utilisateurs. Autrement dit, il est inutile de faire juger l'intérêt d'une fonctionnalité sans en donner l'expérience; le problème est alors celui des questions relatives à de nouvelles caractéristiques dont il n'est pas possible d'expérimenter l'usage.

- la difficulté d'utilisation des commandes est établie à partir des notations fournies aux diverses questions sur la facilité d'utilisation. Ce critère permet de classer les commandes selon le problème qu'elles posent et d'ordonner les dysfonctionnements par une mesure d'insatisfaction. Le sous-ensemble des commandes à problème est constitué de l'intersection des listes établies par les sujets: ces mêmes commandes sont mentionnées dans tous les groupes.

- les *questions ouvertes* permettent de grouper les remarques individuelles en des classes de commentaires (remarques communes à tous les individus). Elles apportent une information non négligeable et enrichissent les questionnaires en évitant les omissions qui se produisent lors de la construction. Cependant ce questionnement est difficilement exploitable lorsqu'il n'y a pas de réponse aux questions de sémantique.

- la vérification de la cohérence intra et inter-sujet est réalisée par la comparaison des réponses fournies aux différents types de question. Une incohérence se traduit par exemple par le fait de ne pas connaître une commande et de juger de son utilité. Ce contrôle de cohérence améliore la qualité des réponses et la fiabilité de l'évaluation.

- l'expérience antérieure des utilisateurs et la méthode d'administration du questionnaire (papier-crayon ou en ligne) ont un effet sur le degré de précision des réponses mais

ne biaisent pas les résultats (par exemple dans la direction d'une critique plus importante des commandes); le questionnaire en ligne est dans la mesure du possible la solution a utiliser.

Bien que la technique soit un moyen satisfaisant d'obtenir des réponses cohérentes, ses limites sont cependant évidentes étant donné qu'elle ne permet pas d'identifier l'organisation de l'activité de l'utilisateur: aucune information n'est fournie sur la structure des procédures de travail, sur la fréquence d'usage des commandes, sur le contexte dans lequel se produisent les erreurs et le temps passé à leur correction. Lorsque l'évaluation doit descendre à ce niveau d'analyse, il n'est plus possible de faire l'économie de données quantifiées; les "mouchards électroniques" fournissent toutes les informations requises.

# 3. Mouchards électroniques

Le recueil automatique des actions mises en jeu par les utilisateurs sur les différents dispositifs de commandes (frappe des touches, événements souris...) est une pratique courante dans les laboratoires (voir par exemple, Hanson et al., 1984; Neals et Simon 1983; Aperley et Field, 1985). L'utilisation de ces techniques en situation naturelle ne va pas de soi du fait que ces contextes sont moins épurés que les situations de laboratoire; la question est alors de savoir dans quelle mesure l'enregistrement automatique d'événements datés, générés par l'utilisateur permet d'inférer les modalités réelles d'utilisation de l'interface.

## *3.1. Problèmes d'utilisation des mouchards en situation naturelle*

Teubner et Vaske (1988) ont abordé cette question pour montrer la faisabilité d'une surveillance de l'utilisation (monitoring) dans des environnements de travail multi-utilisateur. Les difficultés potentielles dans ce contexte tiennent entre autres, à la nécessité de préserver les données recueillies en cas d'interruption système, aux contraintes qui peuvent être créées pour les utilisateurs et/ou à la croissance très rapide des fichiers de données (d'un côté, un enregistrement systématique peut affecter le temps de réponse du système, d'un autre côté un filtrage trop sélectif conduit à perdre de l'information). Les solutions techniques proposées par Teubner et Vaske (1988) pour régler ces questions sont établies autour de compromis entre le niveau de détail des données requises pour l'analyse et la complexité des procédures de filtrage des données intéressantes. Selon eux, les routines logicielles standards (par exemple facturation des services logiciels par la mesure du taux d'utilisation) ne sont par suffisantes pour étudier correctement les interactions, et il faut développer des instruments spécifiques dédiés à la tâche contrôlée.

## *3.2. Construction de logiciels d'enregistrement des événements*

Le développement d'un instrument d'enregistrement automatique opérationnel doit être assuré en trois étapes: l'analyse du problème, l'analyse du logiciel et le plan de traitement des données.

### a) Analyse du problème

Il n'y a pas de limite à la quantité de données que l'on peut recueillir et l'on peut enregistrer des sessions de travail entières. Une analyse du problème doit déterminer quels aspects de l'interaction doivent être observés et quelles informations caractériseront le comportement du système homme-machine que l'on désire étudier. Par exemple, les données peuvent être enregistrées en permanence ou bien uniquement pour identifier les erreurs et les demandes d'aide. Les analyses portent en général sur la *fréquence d'utilisation* des commandes et sur l'identification de *séquences structurées* ("patrons d'usage"): par exemple, Bannon et O'Malley (1985) repèrent des "patrons" d'exploitation du MAN d'UNIX structurés en appels successifs visant à acquérir des informations sur des arguments différents mais fonctionnellement reliés.

Compte tenu de la grande variété des indices qui peuvent être pris en compte, et de la relative facilité du recueil, le point critique de ces techniques est déplacé vers la *spécification précise des objectifs de l'évaluation*, la définition d'*hypothèses* et la *préparation structurée du recueil* et de *l'analyse*. Cette planification décide des événements devant être enregistrés.

b) Analyse du logiciel

L'analyse du logiciel sur lequel est effectué le recueil identifie les composants qui peuvent générer les événements dont on a besoin pour l'étude des interactions. Cette étude technique doit être assurée par un *informaticien expérimenté* pour éviter les difficultés rapportées par Bannon et al (1985). Ces auteurs ont utilisé le monitoring pour identifier les modalités d'utilisation du MAN d'unix et pour évaluer l'impact des aménagements qui ont ensuite été réalisés. Lors de ce contrôle des difficultés d'analyse ont été rencontrées: les données recueillies ne concernent pas uniquement les actions de l'utilisateur mais aussi des informations parasites (appel système à d'autres processeurs) qu'il est difficile de filtrer dans le système considéré. Pour résoudre leur problème d'analyse, Bannon et O'Malley (1985) ont été contraints d'éliminer "à la main" ces parasites avant de pouvoir réaliser un traitement automatique.

c) Plan de traitement des données

Le plan de traitement des données définit les transformations qui doivent être réalisées sur les données brutes pour les rendre exploitables par l'analyste. Ces transformations peuvent consister en des *résumés*, en une extraction des *patrons d'usage* et/ou en une *mise en forme* pour des traitements statistiques. Les entrées clavier ne sont généralement pas suffisantes pour étudier les difficultés, il est difficile de les interpréter et elles doivent être enrichies par le recueil d'autres données. Par exemple, à moins de disposer des informations complémentaires, il n'est pas possible de donner une *interprétation causale des erreurs* et de savoir pourquoi celles-ci se produisent. Des corrélations avec d'autres mesures d'évaluation doivent être réalisées pour la validation et dans la plupart des situations, il est utile d'autoriser des utilisateurs à *commenter l'interaction.*

## *3.3. Faisabilité de l'approche*

La faisabilité de l'approche a été montré par Teubner et Vaske (1988) à partir de deux études: la première concerne l'analyse de l'utilisation de divers sous-systèmes d'une station bureautique intégrée (messagerie, traitement de texte, feuille de calcul...), la seconde illustre comment les mouchards peuvent servir à détecter les dysfonctionnements et à modifier l'IHM d'une messagerie électronique.

a) Etude de l'utilisation d'une station bureautique

Dans la première expérience, les données recueillies concernent:

- la fréquence d'accès à chaque sous-système
- la durée d'utilisation d'une fonctionnalité
- le temps moyen d'utilisation des fonctionnalités au cours d'une session de travail
- le pourcentage du temps total de travail consacré à une fonctionnalité déterminée

Les analyses des données sont réalisées selon deux perspectives destinées à illustrer l'intérêt de l'approche malgré le niveau de généralité des données recueillies. La première analyse décrit l'*utilisation journalière individuelle* sur une période d'une semaine, la seconde différencie les modalités d'*utilisation en fonction de la position hiérarchique* de l'utilisateur dans l'entreprise (gestionnaire, directeur, technicien...).

b) Diagnostic des défauts d'une messagerie électronique

La seconde étude est destinée à identifier les possibilités d'amélioration d'une interface de messagerie électronique. Cette fois, ce sont les interactions entre l'utilisateur et le dispositif qui sont enregistrées. On étudie, sur un échantillon de 152 utilisateurs, la fréquence d'usage d'un ensemble de formulaires et les modalités d'utilisation de leurs champs. Les résultats font apparaître que des *fonctions ne sont presque jamais utilisées* (par exemple, 98% des utilisateurs ne se servent jamais d'une option de filtrage des messages) et que les *formulaires sont mal organisés.* Ces résultats conduisent à des aménagements de l'interface consistant, entre autres, en la création de nouvelles commandes rendant plus explicites les options disponibles et en une modification de la présentation des formulaires.

### *3.4. Avantages de la technique*

Pour les auteurs, les mouchards électroniques permettent d'évaluer l'efficacité d'un dialogue de façon satisfaisante et constituent un outil essentiel de diagnostic des performances des systèmes homme-machine dont les avantages par rapport aux études de laboratoires tiennent à ce que:

- les données recueillies concernent potentiellement des échantillons très larges: elles peuvent théoriquement être acquises auprès de l'ensemble des utilisateurs d'un système.
- cette diversité permet de différencier des groupes d'utilisateur sur la base des stratégies d'utilisation qu'ils développent.
- l'échantillonnage des observations peut être assuré sur une période très longue, ce qui permet de détecter des tendances comportementales et d'analyser les effets de l'apprentissage
- l'installation de nouveaux logiciels peut être étudiée systématiquement de façon empirique: évolution de l'utilisation, appropriation
- la technique n'est pas intrusive

L'intérêt d'un mouchard dépend beaucoup du niveau d'analyse envisagé et du moment où est réalisé l'enregistrement. Par exemple, un instrument identifiant les patrons de commande après la mise à disposition d'un nouveau produit constitue un moyen d'étude écologique de l'évolution de l'utilisation en fonction du temps. Pratiquement cela permet aussi de déterminer à quel moment les utilisateurs sont suffisamment expérimentés pour que les réponses qu'ils fournissent à des enquêtes sur l'utilisabilité du système aient un sens.

# II. Tests de conception

Le développement d'un produit commercialisable constitue généralement une situation assez ouverte: les contraintes initiales peuvent évoluer (modification des caractéristiques de la population d'utilisateurs finaux, échéances de développement, enrichissement des tâches....) et les évaluations concernent des *versions inachevées de l'IHM* dont les fonctionnalités sont instables. Une approche relativement récente suggère que, pour s'adapter à ces fortes exigences, l'évaluation d'un produit doit être réalisée selon un *cycle* correspondant à la *nature itérative du processus de conception.* Dans l'idéal, à chacune des phases du développement, le produit est soumis à une série de tests; les aménagements effectués dans la nouvelle version sont ensuite mis à l'épreuve et ainsi de suite jusqu'à la réalisation d'un produit satisfaisant. Dans cette perspective, on peut distinguer deux classes de questions d'évaluation qui correspondent à des étapes clés du processus de conception:

- *l'évaluation d'alternatives de conception* est réalisée lorsqu'il n'existe aucun critère de choix évident entre plusieurs possibilités. Le recueil de données empiriques doit permettre de hiérarchiser les solutions envisagées. Ces études expérimentales mettent en jeu une méthodologie rigoureuse bien dominée et sont conduites en amont du prototypage,

- le *contrôle de qualité* de l'IHM est assuré soit par une évaluation itérative associant détection des défauts et mesure d'impact des aménagements soit par un banc d'essai du produit fini:

  - la *détection des défauts* est faite par des mises en situation d'utilisation faisant apparaître les éventuelles difficultés d'utilisation liées aux décisions de conception. Les analyses restent locales, focalisées sur des aspects spécifiques de l'interface (qualité des messages d'erreur, complexité des écrans...);

  - les *mesures de l'impact des aménagements* vérifient que les transformations introduites pour réduire les difficultés d'utilisation d'une version précédente sont satisfaisantes;

- les *bancs d'essai* destinés au contrôle de qualité du produit fini sont orientés vers un diagnostic plus global: ils mettent jeu des techniques de recueil sophistiquées et sont parfois réalisés sur des stations d'évaluation.

Le schéma ci-dessous organise ces étapes dans un modèle de l'évaluation de conception que l'on suivra pour illustrer les problèmes méthodologiques qui se posent dans ce contexte:

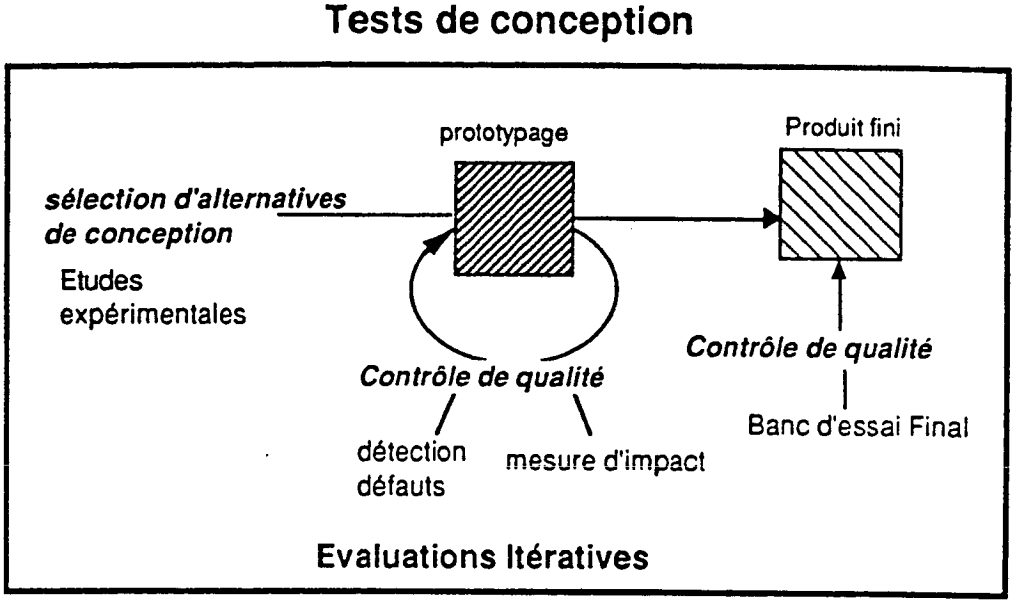



*Fig. 7: Etapes de l'évaluation en cours de conception*

Cette organisation séquentielle reste théorique étant donné que dans la réalité, du fait des contraintes importantes qui pèsent sur le développement, les approches sont souvent moins structurées et les tests ne sont pas systématiques. On discutera ici en particulier des évaluations itératives en les contrastant avec l'approche structurée de l'*ingéniérie de l'évaluation* qui fait de la définition d'objectifs de performance et de la mesure de l'impact des aménagements les déterminants de la qualité des évaluations itératives.

# 1. Sélection d'alternatives de conception: les études expérimentales

Lorsque plusieurs possibilités de conception sont envisagées et qu'aucune d'entre elles ne peut être clairement privilégiée, des études spécifiques doivent déterminer quelle est la solution la plus satisfaisante. Le travail de Bewley et al. (1983) présenté maintenant a été conduit lors du développement du STAR (Smith et al., 1982). Il montre que des études réalisées sans prototype permettent de déterminer, dès les premières phases de la conception, les principes généraux qui guideront le développement ultérieur. Autrement dit, la *qualité d'une évaluation* ne dépend pas de la mise en jeu d'une technologie sophistiquée mais de la précision des objectifs et de la définition de tâches de contrôle adéquates. L'utilisation de protocoles expérimentaux rigoureux conduit de plus à des résultats bien contrôlés.

## *1.1. Illustration: qualités des représentations iconiques*

Ce travail conduit dans la plus pure tradition expérimentale comporte plusieurs études concernant les procédures syntaxiques d'entrée: détermination du nombre et de la sémantique des boutons d'une souris, manipulation directe d'objet graphique.... On mentionnera ici uniquement celles destinées à identifier les propriétés facilitant l'apprentissage, l'identification et la discrimination des icônes. L'évaluation est faite en enregistrant les performances de quatre groupes de sujets, chaque groupe manipulant un ensemble d'icônes élaborées selon des principes différents. Les tests utilisés mélangent les *épreuves formelles et informelles* et présentent une forte redondance qui permet

d'analyser de façon systématique tous les aspects d'une question. Trois types de tests sont utilisés: un test de familiarité , un test de temps de réaction, un test d'opinion.

a) Test de familiarité

Il s'agit d'un test sémiologique visant à déterminer la *capacité d'évocation* des icônes et l'*adéquation de leur dénomination* qui comporte trois épreuves différentes. Dans la première, on présente au sujet une carte représentant une icône et on lui demande d'indiquer ce qu'elle représente. La présentation des icônes est séquentielle et la description est libre. Lorsqu'elle est achevée, le sujet voit l'ensemble des icônes et peut modifier ses descriptions initiales.

La seconde épreuve consiste à montrer au sujet l'ensemble des cartes et à lui fournir le nom d'une icône ainsi qu'une courte description fonctionnelle; le sujet doit sélectionner la carte parmi l'ensemble. La dernière épreuve est une mise en correspondance: le sujet dispose des noms et des cartes et doit les associer.

b) Test de temps de réaction

Ce test comporte une épreuve de *reconnaissance* et une épreuve de *discrimination*. Dans la première, le sujet est placé devant un dispositif affichant les icônes sur un écran. L'expérimentateur propose un nom et le sujet doit indiquer le plus rapidement possible par appui sur une touche (oui-non) si l'icône affichée correspond au nom proposé.

La seconde tâche est une épreuve de discrimination dans laquelle on affiche sur l'écran l'ensemble des icônes et le sujet doit pointer le plus rapidement possible sur celle dont on lui a fourni le nom.

c) Test d'opinion

Il consiste d'abord à demander à la fin des tests aux sujets quelles icônes étaient faciles ou difficiles à sortir du lot. Puis les différents ensembles d'icônes sont présentés aux sujets et ils doivent indiquer leur préférence et extraire "la meilleure" de chaque type.

### *1.2. Limites des approches expérimentales*

Ce travail présente toute la rigueur expérimentale que l'on peut souhaiter et il faut en retenir l'idée d'une *batterie de tests* permettant d'analyser les relations établies entre les symboles et leur référent selon différents points de vue. La redondance introduite autorise un *contrôle de la cohérence* des résultats aux différents tests. Cette recherche s'inscrit dans un courant empirique cherchant à identifier quelles propriétés générales des composants d'une interface améliorent la communication homme-machine (par exemple "quelles propriétés des menus facilitent leur utilisation ?") [1]. Ce type d'approche enrichit les bases de données ergonomiques dans la mesure où les résultats sont généralisables. En l'occurrence, les études présentées ci-dessus ont non seulement permis de sélectionner l'ensemble d'icônes le plus adéquat mais ont surtout mis en évidence l'importance des labels associés aux représentations graphiques et de la qualité du contraste en inversion vidéo.

Cette étude n'est cependant pas totalement satisfaisante pour un ergonome car elle pose une question de *validité écologique* des tâches de laboratoire proposées aux sujets. Ces tâches standards (reconnaissance, discrimination ..) font abstraction du contexte dans lequel les différentes icônes seront présentées. Dans une situation de travail réelle, l'activité de l'utilisateur est pilotée par les *buts* courants et l'individu développe une *attention sélective*: les icônes seront associées à d'autres icônes selon des relations fonctionnelles qui peuvent être déterminantes sur la capacité de discrimination et de détection. Les propriétés perceptives des icônes ne sont pas suffisantes pour les évaluer et la sémantique définie dans le cadre de la tâche qui leur est associée devrait être considérée lors de la sélection d'alternatives de conception.

---

[1] Pour illustrer concrètement, Perlman (1985b) montre que la recherche d'items est plus rapide sur des listes de chiffres que sur des listes alphabétiques, que l'efficacité des mnémoniques (sélecteur) frappés au clavier pour sélectionner un item dépend de la compatibilité du sélecteur avec la liste des items et que les meilleurs performances sont atteintes lorsque les sélecteurs sont les premières lettres des items ou bien des chiffres respectant l'organisation alphabétique de la liste....

# 2. Evaluations Itératives: les tests exploratoires

Les environnements actuels de développement d'interface sont constitués de "kits de construction" (boîtes à outils, gestionnaires de fenêtres, éditeurs d'interface) qui réduisent l'écriture d'un code répétitif et rendent plus faciles les modifications de l'IHM (Fischer et Lemke, 1988). Ces outils de génie logiciel ont popularisé le *prototypage*: il devient possible d'élaborer rapidement des artefacts plus ou moins sophistiqués simulant la sémantique externe de l'interface utilisateur et de tester à moindre coût des alternatives de conception très diverses: localisation des lignes de commande, types de messages ou de prompts devant être utilisés, indentation des données sur l'écran,...

Le schéma de principe en est le suivant:

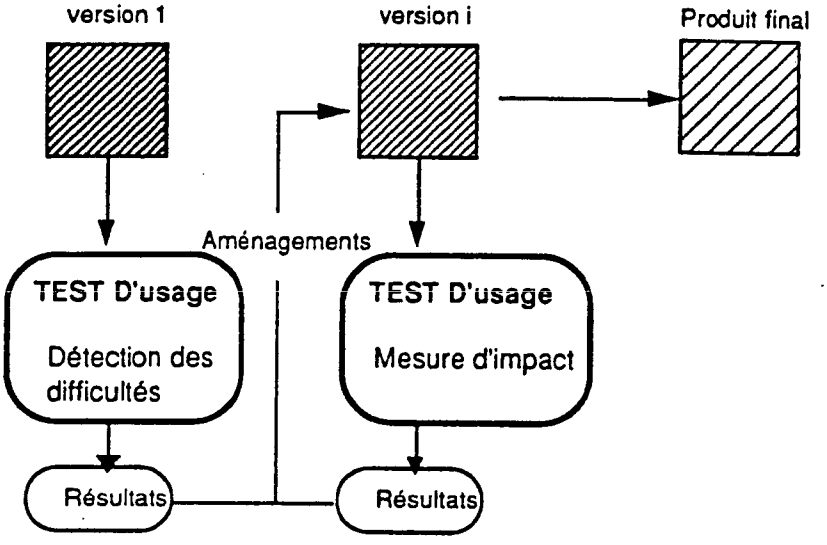



*Fig. 8: Schéma de principe des tests exploratoires*

Ces possibilités ont sensiblement modifié les pratiques évaluatives, mais elles ont eu un effet de bord négatif: les tests d'utilisation conduits au cours de la conception relèvent davantage d'une approche exploratoire que d'une méthodologie expérimentale rigoureuse. C'est ce que l'on illustrera à travers l'analyse de quelques travaux et leur discussion fera ressortir l'intérêt que présente l'approche de l'ingéniérie de l'évaluation.

## *2.1. Avantages théoriques du prototypage*

L'intérêt d'une évaluation de l'interface utilisateur à partir d'une simulation est maintenant couramment admise. Le prototypage est censé *minimiser les coûts* de développement et *optimiser la qualité* de l'IHM. Les transformations de l'interface utilisateur sont en effet peu coûteuses lorsque les principes généraux de séparation du code de l'application et du code de l'IHM ont été appliqués et le développement est d'autant plus rapide qu'une interface peut être réutilisée [1]. De plus, il existe théoriquement une relation directe entre la fréquence des tests et la qualité de l'IHM: les évaluations itératives ont pour effet de bord d'installer progressivement des *standards de présentation* et de comportement ayant été validés et permettant à terme d'obtenir plus rapidement des versions satisfaisantes. Enfin un avantage non négligeable est que l'*implication des utilisateurs* dans la conception limite les rejets de l'outil informatique. On discutera ici deux avantages reconnus du prototypage: le réalisme des simulations et l'efficacité des tests (Bury, 1985; Lund, 1985).

---

[1] La "rentabilité" réelle du prototypage dépend cependant largement de la compatibilité de l'environnement de développement du prototype et de celui de l'application: même si tout prototype est censé pouvoir être "jeté", il serait souhaitable de pouvoir implémenter le produit final à partir du code d'une version reconnue satisfaisante.

a) Réalisme des simulations

Le réalisme de la mise en situation sur prototype tient à ce que l'utilisateur est confronté au dispositif et à l'environnement d'assistance (en ligne ou documentaire) qui seront effectivement utilisés sur le poste de travail. Les données recueillies sont censées avoir de ce fait une validité plus importante que lors d'analyses hors contexte. De plus, l'observation des *réactions spontanées* de l'utilisateur par les concepteurs doit théoriquement leur permettre de prendre un recul nécessaire: il est plus facile d'admettre l'existence d'un problème lorsqu'on en a une démonstration concrète que lorsque celui-ci est présenté à travers un tableau de chiffres dans un compte rendu d'expert. Cette *fonction de communication* qu'assure le prototypage prend une importance croissante étant donné que la visualisation directe des alternatives de conception évite aussi les malentendus classiques qui existent entre concepteur et utilisateur lorsqu'un produit n'est défini que par des documents textuels (cahier des charges, spécifications fonctionnelles,....).

b) Efficacité des tests sur prototype

Les évaluations sur prototype permettent la *détection précoce des problèmes potentiels*: les difficultés de communication homme-machine font apparaître les contraintes introduites par le dispositif sur "la façon de penser" de l'utilisateur. Les données recueillies sont très précises (utilisation des commandes, calcul de fréquence, contexte d'usage...) et supportent le *diagnostic causal*, i.e. l'identification de ce qui conduit l'utilisateur dans une direction erronée. Selon que l'origine de l'erreur est liée à l'intitulé d'une commande, au texte d'un message ou à la présentation d'un concept en des termes non familiers, des solutions spécifiques doivent être recherchées.

La simulation offre non seulement aux utilisateurs la possibilité d'exprimer leurs *préférences* mais elle a aussi des retombées directes sur la *conception de l'aide* à l'utilisateur (documentation et formation). Par exemple, l'identification des concepts clés pour les utilisateurs permet de hiérarchiser les notions, de structurer le processus d'apprentissage et de limiter la taille de l'environnement documentaire.

Cette présentation peut laisser supposer que les études conduites sur prototype supportent de façon satisfaisante les tests de conception. Cependant, dans le meilleur des cas, l'utilisation d'une technologie sophistiquée améliore le contrôle expérimental et la précision des données recueillies, mais elle n'est jamais suffisante pour déterminer la qualité d'une évaluation, celle-ci dépendant de la méthodologie mise en oeuvre. C'est que l'on peut montrer en examinant plus en détail les limites des simulations pour en faire ressortir les avantages réels du protoptypage.

## *2.2. Limites des simulations et avantages réels du prototypage*

La structure générale d'un cycle d'évaluation sur prototype est décrite dans le schéma suivant:

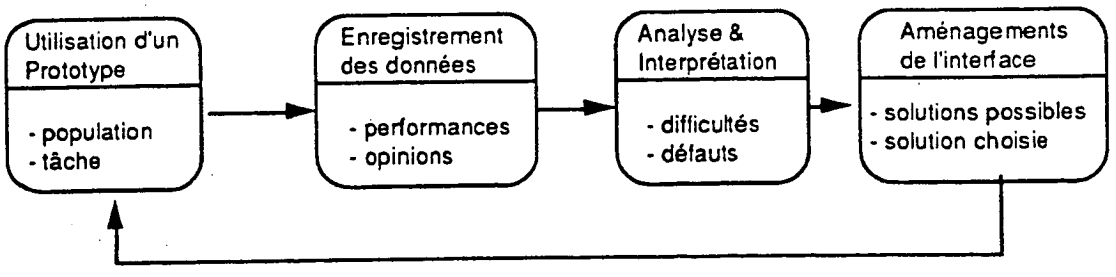



*Fig. 9: Cycle d'évaluation sur prototype*

Selon ce schéma, le réalisme d'une simulation dépend avant tout du *réalisme des tâches* et de la *représentativité de la population* testée. Il faut y rajouter d'autres facteurs plus fondamentaux car, toutes les simulations comportent des limites intrinsèques: aussi réalistes qu'elles s'efforcent d'être, elles ne reproduisent jamais parfaitement l'environnement et les conditions de travail réel; des facteurs tels que le stress, la motivation peuvent difficilement être étudiés *in vitro*. Or, la motivation est un facteur qui permet de dépasser tous les défauts d'un dispositif, ce qui en l'occurrence peut biaiser considérablement les évaluations: les futurs utilisateurs peuvent manifester un intérêt sensible au cours des enquêtes et s'acharner sur les tâches qu'on leur propose

en laboratoire, sans pour autant que cela garantissent l'exploitation du dispositif lorsqu'il sera installé sur le poste de travail. Pour déterminer dans quelle mesure ils se serviront effectivement du produit lorsqu'il sera disponible, des analyses de besoin conduites en situation naturelle doivent être réalisées. Ceci étant, les diverses conditions de "validité écologique" d'un test ne sont pas toujours satisfaites pour de multiples raisons analysées ci-dessous.

a) Réalisme et cohérence inter-application

Un logiciel est souvent testé de façon isolée alors qu'en situation naturelle l'utilisateur peut être plongé dans un environnement technologiquement saturé. La compatibilité entre les divers dispositifs (ou entre versions successives d'un logiciel) est une dimension essentielle de l'utilisabilité qui devrait être prise en compte. Par exemple, une interface utilisateur peut obtenir un score satisfaisant sur un banc d'essai et cependant être difficilement utilisable du fait qu'elle nécessite des procédures de commandes incompatibles avec celles des autres dispositifs (inversion d'opérations dans une séquence de commandes, identité de labels...). La détection de ces difficultés de *transfert de procédures* suppose un échantillonnage établi à partir d'études de terrain identifiant le contexte plus général d'utilisation du dispositif.

b) Réalisme et conditions de passation

Selon les techniques de recueil utilisées, des biais supplémentaires risquent d'être introduits, comme c'est par exemple le cas avec la *verbalisation simultanée*.

Les nombreuses critiques émises à l'encontre de l'utilisation de cette technique pour la modélisation cognitive [1] ne s'appliquent pas aux situations d'évaluation du fait que le niveau des analyses qui y sont conduites est pour le moment largement en deçà des exigences de modélisation: par exemple, la version dégradée utilisée par Lund (1985) lors de l'évaluation de l'interface d'un système d'aide à la modélisation ne donne pas lieu à la transcription de protocoles; il s'agit davantage d'une *observation commentée* que d'un véritable "raisonnement à voix haute"[2]. Lund (1985) illustre en quoi ces techniques introduisent du "bruit" dans l'évaluation: les sujets réalisent une tâche simple en commentant à voix haute l'interaction mais de ce fait, tout chronométrage est rendu impossible. Le "protocole verbal d'évaluation" permet cependant de recueillir des informations sur l'ensemble du processus d'interaction et décrit en détail le point de vue du sujet sur les caractéristiques particulières de l'interface. Le commentaire identifie en temps réel la nature des difficultés de l'utilisateur et donne une idée des questions qu'il se pose, mais n'est vraiment exploitable que lors des phases exploratoires.

---

[1] Elles portent sur des questions de réalisme psychologique, de biais liés à la verbalisation et de validité pour inférer les processus sous-jacents (voir Ericsson et al., 1984 pour une discussion de ce sujet).

[2] Dans sa version plus technique, l'utilisation de protocoles verbaux nécessite un apprentissage important du fait de difficultés de recueil et d'analyse. Lors du recueil, les intrusions provoquant l'explicitation par le sujet ne doivent pas introduire de biais de questionnement. L'analyse des protocoles est généralement faite en trois temps (transcription textuelle des enregistrements , segmentation et définition d'unités d'analyse, codage dans un vocabulaire plus réduit). Lorsque le niveau de détail requis est important, l'analyse devient très complexe et doit être assurée par des spécialistes. Un traitement complet validé nécessite plusieurs analystes travaillant de façon indépendante selon une procédure de notation bien établie. Le rapport de durée recueil/ traitement d'une analyse détaillée varie de 3 à 10 selon les auteurs (Sweeney & al., 1987) et des techniques d'analyse automatisée sont en cours de développement (Bailey, & al., 1987).

c) Simplification des tâches effectuées

Dans quelle mesure les tâches choisies pour l'évaluation sont-elles représentatives de celles qui seront mises en jeu dans la situation naturelle ? Les tests proposent en général de traiter des problèmes qui ont peu de choses à voir avec la complexité du travail, avec le flou de certaines situations et l'enchevêtrement des tâches que devra assurer l'utilisateur. Un argument donné par Lund (1985) pour justifier l'utilisation d'un tâche simple est que "...l'objectif est de tester l'interface et non les compétences de l'ingénieur". Une évaluation envisagée dans cette perspective ne peut concerner que des aspects de surface (codage, compréhension des labels, des textes) et ne permet pas de traiter les questions d'*adéquation à la tâche*: il n'est pas possible de prédire les difficultés ultérieures rencontrées lorsque le dispositif sera utilisé pour traiter des problèmes complexes. Par exemple, la récupération d'un contexte après les interruptions qui se produisent en situation de travail est un aspect critique de nombreux logiciels: l'existence d'une fonction gérant un historique des commandes est une facilité appréciable mais dont l'intérêt peut difficilement apparaître au cours d'une évaluation standard.

d) Evaluation ponctuelle et apprentissage

Un autre aspect de la simplification tient à ce que la dimension d'*apprentissage* est souvent négligé. Un premier point est que chaque cycle d'évaluation pose des questions spécifiques: les premières situations d'utilisation sont particulièrement critiques car elles font apparaître des difficultés avant que les sujets n'aient développé des solutions permettant de les éviter. De plus, les véritables difficultés ne pourront apparaître qu'après une utilisation intensive que le prototypage ne permet pas de simuler. Que dire alors de résultats établis en quelques heures par rapport à des situations d'interaction qui occupent des semaines ? La fréquence d'utilisation et la nature des fonctions utilisées par une population donnée ne peuvent être dérivées d'une expérience qui ne dure qu'une heure (Teubner et al. , 1988). Certains facteurs qui paraissent négatifs sur le court terme peuvent avoir un effet positif à long terme. Par exemple les facilités d'expression des "préférences" complexifient la compréhension du débutant, mais elles améliorent sensiblement le confort d'utilisation pour l'opérateur expérimenté. Une solution est alors de poursuivre l'évaluation du système après qu'il a été mis à disposition des utilisateurs finaux (beta-test) en installant par exemple un "cahier de doléances" tenu par les utilisateurs et recensant systématiquement les difficultés qu'ils rencontrent.

e) Apprentissage guidé et découverte des dispositifs

Dans la plupart des évaluations sur prototype, les sujets doivent d'abord réaliser l'apprentissage de l'utilisation à partir d'une documentation ou en étant assisté, puis réaliser une tâche dans laquelle ils sont guidés pas à pas avant d'effectuer le test expérimental proprement dit. Quiconque a déjà procédé à des observations en situation naturelle sait que les sujets ne procèdent pas de cette façon: ils ne lisent pas les manuels, ils ont du mal à suivre les didacticiels et ne poursuivent qu'un but: *réaliser leur tâche avec le système le plus rapidement possible.*

Le guidage complet au cours des évaluations passe à côté d'une des raisons de la sous-utilisation: dans bien des cas celle-ci tient à ce que les utilisateurs ne savent pas quel est l'outil ou la fonction qui peut servir à régler le problème qu'ils se posent. Pour le découvrir, ii faudrait qu'il puissent établir la relation entre l'outil et le contexte du problème mais celle-ci est toujours pré-définie dans les situations expérimentales.

f) Représentativité de la population

La sélection de sujets représentatifs de leur population parente n'est pas une véritable difficulté: des procédures d'échantillonnage permettent de traiter correctement cette question. On sait, par exemple, qu'il est vain d'espérer cerner complètement les problèmes d'une IHM par l'observation d'un seul utilisateur car il existe d'importantes *différences inter-individuelles*: s'il est possible d'identifier un "noyau dur" de difficultés rencontrées par tous les utilisateurs (20 à 40% de recoupement d'un individu à l'autre), chaque individu rencontre des problèmes spécifiques (Hammond et al., 1985). La constitution d'un groupe d'utilisateurs aussi proche que possible des utilisateurs finaux du produit suppose d'avoir une description précise de la *population cible* (formation, niveau d'étude, expérience de l'informatique, de la tâche, de la frappe dactylographique...) et de connaître

la *nature des tâches.* Les difficultés les plus fréquentes et la gamme des dysfonctionnements plus rares peuvent être identifiées avec un groupe d'utilisateurs novices peu nombreux.

Le véritable problème tient à ce qu'il arrive qu'un logiciel destiné à l'origine à une population bien déterminée se retrouve utilisé par une autre population ayant des caractéristiques très différentes. C'est par exemple le cas lors de l'ouverture d'un nouveau marché commercial, ou lors de l'enrichissement des tâches d'une catégorie professionnelle dans une entreprise. L'évaluation réalisée sur la population initiale n'est d'aucun secours pour prédire les difficultés de la nouvelle population. Il est difficile de prendre en compte ces aspects dans une expérimentation planifiée, mais cela suggère néanmoins d'enrichir les échantillonnages par le recours à des *utilisateurs non ciblés* par le produit.

g) Efficacité des tests et exploitation des données

Les techniques de recueil de données sont souvent assez sophistiquées (enregistrement vidéo des actions de l'utilisateur, monitoring des interactions, enregistrement des verbalisations... ) et les variables recueillies sont très nombreuses. Par exemple, les données enregistrées pour évaluer la qualité de messages d'erreur (Isa et al., 1983) sont les suivantes:

- le type d'erreurs commises
- la fréquence d'erreur
- la fréquence d'utilisation de l'aide ou du manuel de référence (ces données servent à identifier la qualité des messages d'erreur en les classant)
- le temps passé dans le contexte d'aide
- le nombre de lignes éditées
- le temps de raisonnement
- le temps d'édition
- la durée de la session
- les réactions de sujets aux messages d'erreur
- les commentaires et l'opinion relatifs au messages
- une reformulation des messages

Dans la plupart des travaux consacrés au contrôle de qualité, malgré le nombre impressionnant de variables enregistrées, les données ne font pas l'objet de traitements statistiques élaborés (par exemple, Neal et Simon, 1983 ne fournissent aucune indication précise sur le traitement des variables enregistrées). Typiquement dans Isa et al. (1983), les tests statistiques utilisés établissent des corrélations entre la durée de résolution de problème, le nombre d'erreurs commises et la qualité des messages affichés mais ils ne définissent pas les déterminants de cette qualité et c'est l'information qualitative fournie par le sujet qui présente le plus d'intérêt. Par exemple, les *reformulations* des messages d'erreurs proposées par les sujets ont une portée générale: elles dépassent le simple aménagement local et il est possible d'en dériver des règles d'amélioration de certaines classes de messages. Le point important de l'évaluation est justement dans l'abstraction de règles de qualité d'un message et dans la compréhension de ce qui fait son adéquation.

## *2.3. Ingéniérie de l'évaluation*

Les études exploratoires qui ont été mentionnées comportent beaucoup des défauts de l'empirisme à outrance: absence d'hypothèse, traitement incomplet des données recueillies, portée locale ... L'ingéniérie de l'évaluation s'efforce de dépasser ces limites et d'obtenir un meilleur contrôle de l'analyse en introduisant des contraintes de spécification. Deux principes de base sont utilisés:

- l'établissement d'une situation de référence par la *spécification de performances d'usage*
- la mise au point de techniques d'*analyse de l'impact* hiérarchisant les solutions d'amélioration d'une interface

a) Spécification de performances d'usage

L'approche repose sur un principe de conception par objectif très classique en ingéniérie (cf. par exemple Carroll et Thomas, 1985; Hewett et Meadow, 1986). L'aspect original tient à ce que ce sont des *performances d'usage* qui sont prises en considération et qui sont établies dès la première phase de spécification du produit. Le schéma de principe est le suivant:

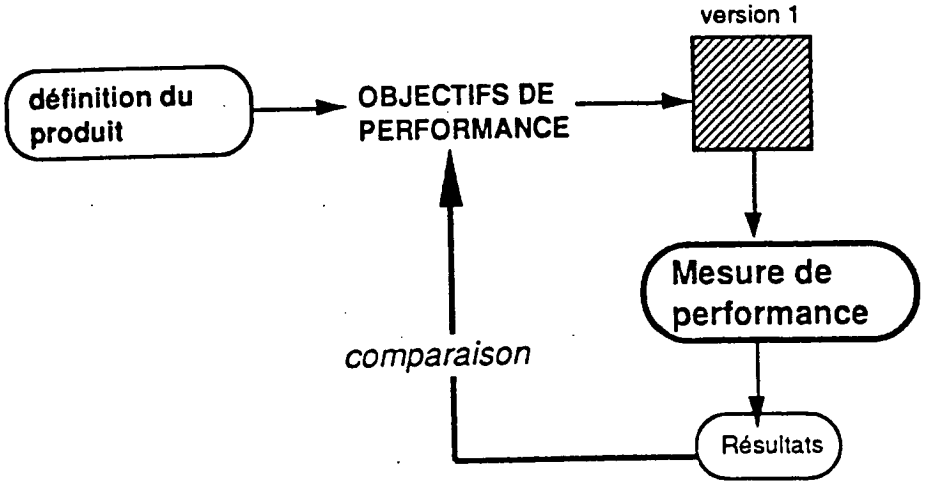



*Fig. 10: Ingéniérie de l'évaluation: spécification de performances d'usage*

La méthodologie consiste à identifier par anticipation les caractéristiques que devra avoir le produit, à les quantifier et à vérifier dans quelles proportions les objectifs attendus ont été atteints. Trois étapes principales ont été énoncées par Butler (1985) :

- les objectifs du produit sont établis lors de la définition du logiciel par un ensemble d'attributs relatifs à sa finalité: fonctionnalités, spécifications techniques, capacités offertes à l'utilisateur. Parmi celles-ci, la facilité d'apprentissage constitue un attribut clé.

- la seconde étape détermine comment les fonctions du système peuvent être implémentées d'une façon qui satisfasse les exigences ergonomiques. Elle exploite les principes généraux d'ergonomie et de psychologie.

- l'évaluation empirique vérifie ensuite que le produit satisfait les objectifs définis.

b) Illustration

Butler (1985) illustre cette démarche en rendant compte du développement d'un logiciel destiné à des analystes financiers et dont la *facilité d'utilisation* déterminera la réussite commerciale. Ce critère est opérationalisé en posant qu'un utilisateur qualifié doit pouvoir résoudre un problème élémentaire en 180 minutes par auto-instruction [1].

Les objectifs à satisfaire sont définis par un critère statistique (l'intervalle de confiance de la moyenne des temps de résolution doit être inférieur à la durée maximale, soit 180 minutes) et par une contrainte de performance (un utilisateur doit pouvoir se sortir tout seul d'une situation d'erreur). Les tests de performance conduits consistent à résoudre un problème d'entraînement,

---

1 La qualification d'un utilisateur est définie par:

- la familiarité avec les concepts du domaine financier
- la capacité à organiser les problèmes de ce domaine
- une expérience de l'utilisation d'un clavier
- une familiarité avec le matériel utilisé.

puis à traiter trois problèmes tests à l'aide du seul manuel d'utilisation du logiciel[1] . Selon l'auteur, l'approche présente plusieurs avantages:

- la spécification d'objectifs de conception et de développement est un concept familier pour les ingénieurs qu'il est facile d'étendre en lui adjoignant les performances de l'utilisateur;
- elle génère chez les concepteurs une recherche d'avis autorisés auprès des spécialistes en facteurs humains;
- elle rend l'évaluation moins discutable puisqu'une situation de référence est définie dès le départ.

On peut y rajouter qu'elle contraint l'ergonome à énoncer précisément ses attentes, i.e. à définir de façon opérationnelle les critères de facilité d'apprentissage et d'utilisation.

c) Mesure d'impact dans les évaluations itératives

Dans les évaluations itératives usuelles, les résultats des tests d'usage conduisent à de nouvelles versions de l'interface et des "mesures d'impact" vérifient l'effet des aménagements introduits. Cette vérification est parfois inscrite dans un cycle de développement régulier comme dans Bewley et al. (1983) mais elle n'est pas systématiquement réalisée et se limite souvent à un contrôle expérimental ponctuel (Lund, 1985 ou Bannon et O'Malley, 1985) dont l'objectif est de déterminer dans quelle mesure les procédures d'utilisation habituelles évoluent après les modifications introduites[2]. Le schéma suivant décrit cette structure classique du cycle de développement itératif .

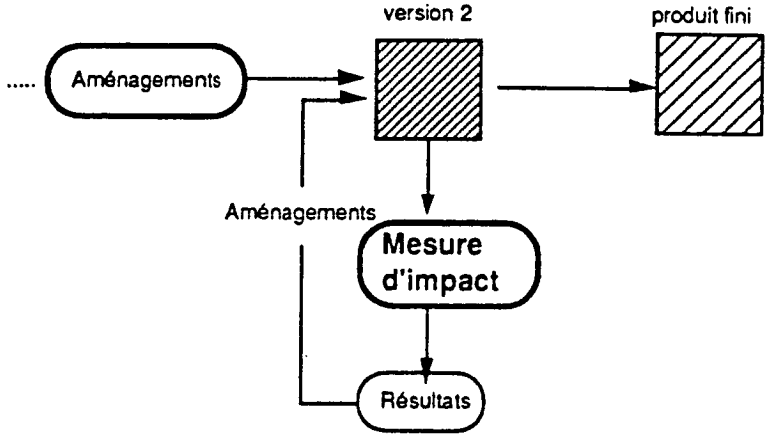



*Fig. 11: Principe des évaluations itératives*

Ces mesures d'impact s'inscrivent dans la même démarche empirique que les tests exploratoires: l'analyse est réalisée *a posteriori* et la conception relève de l'essai-erreur. Il serait intéressant de

---

1 Les problèmes utilisés pour les tests satisfont trois critères:

- ils sont représentatifs des problèmes de modélisation financière que des sujets novices voudraient traiter avec le logiciel;
- ils sont suffisamment simples pour être traités dans les délais imposés, mais ne sont pas triviaux;
- ils nécessitent une mise en oeuvre représentative des fonctionnalités et des caractéristiques de l'interface.

2 Bannon et al (1985) par exemple vérifient l'adéquation d'un logiciel documentaire facilitant la consultation rapide. Le principe consiste à comparer la fréquence d'utilisation du nouveau service de consultation avec celle du système d'aide classique (MAN d'UNIX) avant et après l'implémentation du service. Le recueil des données est réalisé par monitoring et un formulaire en ligne permet aux sujets de commenter la qualité du service qui leur a été apporté.

pouvoir hiérarchiser les difficultés d'utilisation rencontrées et d'estimer *a priori* l'intérêt d'une solution au regard d'une autre. Good et al. (1986) ont apporté une contribution importante à l'ingénierie de l'évaluation en introduisant une technique de sélection des solutions d'amélioration à partir de ce qu'il appellent **l'analyse de l'impact.**

d) Analyse de l'impact des solutions de conception

Good et al. (1986) ont appliqué la méthodologie des performances d'usage pour améliorer l'interface du gestionnaire de fenêtres d'une station de travail. L'utilisabilité du produit est définie par des métriques rendant compte de la performance initiale de l'utilisateur sur un ensemble de tâches et des appréciations subjectives du système après que l'utilisateur a réalisé les tâches (opinion initiale). Les niveaux attendus d'utilisabilité sont définis de façon précise et les auteurs distinguent, de surcroît, les cas les plus favorables et les plus défavorables qu'ils peuvent rencontrer. Le tableau ci-dessous résume leurs attentes.

| Attribut | Technique de mesure | Métrique | Cas défavorable | Niveau attendu | Cas favorable | Niveau actuel |
|---|---|---|---|---|---|---|
| Performance initiale | banc d'essai de tâches | vitesse de travail | idem V1 | 20% > V1 | V1* 3 | idem V1 |
| Opinion initiale | Questionnaire d'attitude | différentiel sémantique | 0 | .25 | 1 | -0.5-0.5 |

V1 = version 1 de l'interface

*Tab. 1: Analyse de l'impact des solutions (d'après Good et al., 1986)*

Le problème traité par Good et al. (1986) est de pouvoir choisir parmi les solutions d'amélioration de l'interface initiale (V1) celles qui permettront d'atteindre les objectifs de performance d'usage [1]. La technique proposée pour sélectionner la solution la plus efficace repose sur les comportements observés lors du premier test d'utilisation d'un prototype et analyse l'impact des solutions de conception.

e) Sélection des solutions d'amélioration de la conception

L'analyse est réalisée en 4 temps:

- mesure du niveau actuel d'utilisabilité

Des ingénieurs ayant une expérience du système mais non celle de la souris réalisent des tâches simples de manipulation de fenêtres: créer des fenêtres, coupler le clavier aux fenêtres, les déplacer ... Les variables dépendantes enregistrées concernent la vitesse de travail (pourcentage de tâches du test réalisées par tranches de 3 minutes )[2] et les évaluations subjectives (questionnaire d'attitude).

- analyse de l'origine des difficultés

le diagnostic des difficultés est réalisé par des analyses successives d'enregistrements vidéo des sessions de travail. Dans une première lecture, on identifie les tâches élémentaires réalisées; une seconde lecture donne une estimation du temps passé dans chacun des problèmes.

---

1 NB: il ne s'agit pas de choisir la meilleure mais bien celles qui suffisent à satisfaire les objectifs.

2 Temps mis par un opérateur expérimenté pour réaliser la tâche.

- prédiction des possibilités d'amélioration de l'utilisabilité

Une estimation du temps gagné par la résolution des problèmes est effectuée en soustrayant le temps total des problèmes à la durée de réalisation de la tâche. Une nouvelle estimation de la vitesse de travail est basée sur le temps réduit (on suppose l'indépendance entre les différents problèmes et leur résolution).

- classement par rang des difficultés

La classification par rang des difficultés est faite en fonction de l'impact qu'aura leur résolution sur l'augmentation de la vitesse de travail.

f) Choix des solutions: compromis entre facilité d'implémentation et efficacité

Les résultats de l'analyse d'impact sont présentés à un groupe d'ingénieurs qui discute les mérites des diverses solutions envisageables. Un des aspects intéressants est que tous les problèmes identifiés ne sont pas traités et seuls certains des plus importants sont pris en compte. Les choix sont effectués en estimant le coût des solutions par rapport à leur impact et les *compromis* établis par les développeurs positionnent la facilité d'utilisation par rapport à la facilité d'implémentation.

La conception et l'évaluation reposent davantage sur l'établissement de compromis que sur la recherche de solutions optimales. Dans cette perspective, les aspects positifs et négatifs doivent être considérés en fonction d'un contexte et la solution d'un problème d'évaluation doit être envisagée au regard de caractéristiques externes. Par exemple, la complexité d'un langage de commande (estimée par la taille du lexique et ses capacités d'extension) est un handicap si l'interface doit être utilisée occasionnellement par des individus peu expérimentés mais devient un atout si la population doit acquérir une expertise pour atteindre le niveau de performance requis dans le cadre du travail.

La démarche générale de l'ingéniérie de l'évaluation est résumée dans le schéma suivant:

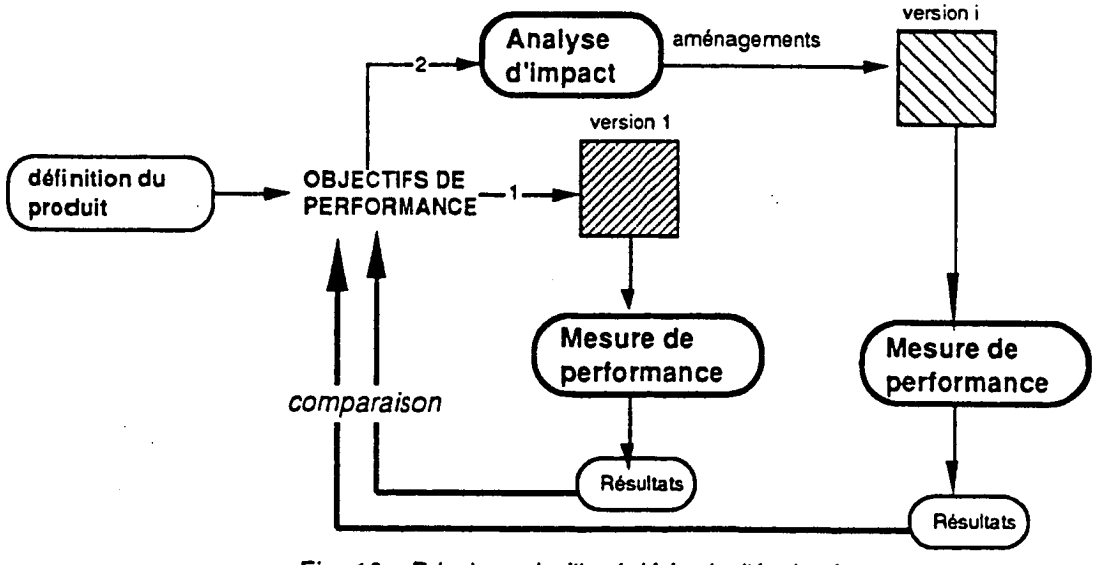



*Fig. 12: Principes de l'ingéniérie de l'évaluation*

# 3. Contrôle de qualité: bancs d'essai de produit fini

Contrairement aux tests de conception qui se focalisent sur un aspect particulier de l'interface (syntaxe d'entrée, syntaxe de sortie, qualité des messages...) les mesures d'"utilisabilité" d'un contrôle final doivent constituer une **évaluation globale du produit** considéré (logiciel et/ou documentation) et doivent en faire apparaître les avantages et les inconvénients. Une difficulté est alors d'identifier les variables critiques qui caractériseront la facilité d'usage et d'utilisation du dispositif et d'élaborer le banc d'essai de tâches supportant l'évaluation. Pour limiter ces difficultés, deux approches complémentaires ont été engagées: l'une portant sur l'établissement d'une méthodologie "universelle" de contrôle de qualité, l'autre mettant l'accent sur l'environnement matériel facilitant le diagnostic ergonomique des interfaces utilisateur. De véritables *stations d'évaluation* regroupant des installations techniques dédiées au recueil de données empiriques ont été mises au point dans certains centres de recherche [1]. Elles comportent en général des moyens d'observation élaborés (glace sans tain, caméras, microphones...) et permettent parfois un "play-back" reproduisant la dynamique du dialogue interactif réalisé au cours de la session (on pourra en trouver une illustration pour l'évaluation de la qualité de messages d'erreur dans Isa et al. , 1983). Malgré cette technologie sophistiquée, ces études présentent les mêmes limites que les tests exploratoires conduits sur prototype. En voici une illustration rapide à partir du travail de Neal et Simon (1983) consacré à l'évaluation finale d'un logiciel et de sa documentation.

## *3.1. Station d'évaluation: illustration*

a) Contexte

Dans l'environnement utilisé par Neal et Simon (1983), le sujet réalise un ensemble de tâches pré-enregistrées à partir d'un terminal classique. Il peut demander à un expérimentateur situé à l'extérieur une aide spécifique qui lui est affichée sur un écran vidéo auxiliaire. Le poste de travail de l'observateur est situé dans une salle extérieure et comporte deux moniteurs vidéo, l'un présentant une image générale de l'utilisateur à son poste de travail, l'autre présentant l'image de la documentation utilisée par le sujet. La méthodologie est la suivante: avant toute évaluation, le sujet subit une épreuve d'apprentissage au cours de laquelle on enregistre le temps mis pour acquérir le programme d'apprentissage, le temps mis pour atteindre un niveau de performance déterminé et les difficultés rencontrées. Sur ce dernier point, les indicateurs objectifs utilisés concernent:

- la fréquence d'affichage des messages d'erreur
- l'incapacité du sujet à trouver l'information recherchée
- la fréquence d'usage de la fonction d'aide
- le temps total passé dans le contexte d'aide
- le recours à une aide extérieure
- l'efficacité d'utilisation du dispositif (procédures sous-optimales)

Lors du test, l'activité des sujets est enregistrée de façon très précise: chaque entrée au clavier est transmise au calculateur de contrôle qui assure un pointage chronométrique continu, cumulant le temps depuis le début de la session de test. Voici à titre d'illustration la liste des variables enregistrées pour étudier la facilité d'utilisation [2]:

---

1 La station du "Human Factors Center" d'IBM de San José a par exemple été utilisée pour mettre au point un matériel d'apprentissage d'éditeurs de texte (Clauer, 1982) , pour étudier l'interrogation de bases de données (Ogden & al., 1982), et pour développer le Basic IBM (Bury, 1983).

2 D'autres variables non standard sont par ailleurs enregistrées au cours de la passation:
- délai entre le début de session et la première entrée clavier
- durée totale de la session
- temps cumulé dans le système d'aide
- fréquence d'utilisation de chaque fonction
- nombre d'appel à l'aide
- fréquence d'utilisation des commandes

- temps mis pour réaliser des tâches sélectionnées
- succès ou échec dans la tâche
- fréquence d'usage des commandes et des caractéristiques du langage
- temps passé dans la documentation
- commentaires, suggestions, préférences

En fin de session, le sujet presse une touche spéciale qui interrompt le chronométrage.

La station d'évaluation a servi à analyser l'usage d'une documentation i.e. à "identifier la facilité avec laquelle l'individu peut retrouver l'information qu'il recherche". Cet objectif reste très général et l'on peut facilement imaginer de nombreuses raisons pour lesquelles il est difficile de "trouver l'information que l'on recherche": représentation inadéquate du problème initial, indexation de l'information selon des codes arbitraires.... L'exploitation d'une documentation, l'accès aux informations requises nécessitent une activité cognitive importante et l'on ne peut cerner les difficultés rencontrées qu'en engageant des études précises (de préférence cliniques) de ces activités. A ce titre, le travail de Neal et Simon pose une intéressante question d'adéquation des techniques de recueil aux objectifs poursuivis. Cette question est discutée ci-dessous en considérant les moyens audo-visuels, fréquemment utilisés lors des évaluations empiriques (Neal et Simon, 1983; Lund, 1985; Bewley et al. 1983; Good et al. , 1986).

b) Protocoles vidéo et analyse cognitive

Dans l'exemple rapporté ci-dessus, le contrôle de l'utilisation de la documentation est réalisé à partir d'un moniteur vidéo: l'observateur doit saisir en temps réel des commentaires, des mnémoniques caractérisant les activités de l'utilisateur, et les bandes vidéo sont ensuite analysées en "play-back". Les codes saisis correspondent par exemple aux observations suivantes:

- utilise l'index du document
- explore le chapitre 3
- lit la page 42
- regarde la page 19 pendant la frappe

Vouloir évaluer la qualité d'une documentation papier à partir d'enregistrements magnétoscopiques paraît être une gageure: on peut se demander en effet dans quelle mesure les observables sont pertinents pour identifier des difficultés de lecture. Les données que l'on va recueillir seront *macroscopiques* (temps de lecture d'une page, nombre de pages tournées, parcours de consultation...) et répondront à des questions générales. Dès que les exigences de l'analyse deviennent plus importantes et que l'on veut connaître par exemple la nature exacte de l'*information traitée* au cours de la lecture ou les *difficultés de compréhension* qui sont aussi des aspects essentiels de la qualité d'une documentation, il faut enregistrer des données plus adéquates (mouvements oculaires, trace et durée des fixations...) et procéder à des études (expérimentales ou cliniques) axées sur des tâches de compréhension.

Autrement dit, le recours aux images n'est intéressant que dans la mesure où les données observables sont pertinentes par rapport au problème posé et que par exemple la gestuelle occupe une place importante dans le contexte. Bewley et al (1983) utilisent justement ce matériel pour identifier les difficultés d'utilisation de l'interface graphique du Star à partir d'une tâche expérimentale de création et de modification des dessins. L'enregistrement vidéo des écrans et du sujet permet d'analyser les nombreuses manipulations graphiques très difficiles à suivre en temps réel.

c) Intérêt des protocoles vidéo dans les évaluations d'IHM

Outre le fait que la technique *évite les biais* liés à la présence d'un observateur, comme tout enregistrement, elle *préserve des événements* que la seule observation est souvent incapable d'appréhender en temps réel. La *reproductibilité des analyses* permet d'en affiner le niveau et de vérifier précisément les mesures expérimentales. Par exemple, Good et al. (1986) et Bewley et al (1983) réalisent leurs analyses en deux temps: un premier traitement des sessions de travail enregistrées a pour but d'identifier les difficultés des utilisateurs. Une seconde lecture permet dans le premier cas d'acquérir une estimation du temps requis pour résoudre chacun des problèmes et dans le second cas, de classer les incidents critiques selon qu'ils traduisent un problème avec le

prototype, avec l'interface, avec la situation d'apprentissage ou avec la procédure expérimentale utilisée.

Il est cependant fréquent que les données acquises par des moyens audio-visuels (enregistrement des écrans) ne soient pas exploitées: par exemple Lund (1985) signale avec une naïveté désarmante que les résultats qu'elle présente auraient pu être obtenus sans magnétoscope. L'absence d'analyse traduit les difficultés actuelles de traitement: par exemple, l'interprétation correcte d'écrans de travail enregistré en continu nécessite de connaître le contexte des événements et il est souvent nécessaire de lire complètement la bande (Lund, 1985). Il ne semble pas encore y avoir de grilles d'analyse des données vidéo équivalentes à ce qui existe déjà pour les mouvements oculaires et utilisables pour l'évaluation.

La vidéo assure principalement pour l'instant une *fonction de communication* (Lund,1985; Bewley et al., 1983): c'est un outil de démonstration qui associe au "poids des mots le choc des images": les développeurs deviennent plus sensibles aux difficultés des utilisateurs lorsqu'ils en ont la visualisation directe.

### *3.2. Méthodologies d'évaluation*

L'approche empirique du contrôle de qualité est actuellement en cours de structuration, c'est ce qui transparait des travaux récents présentés dans la littérature: les techniques développées dans le cadre de l'ingéniérie de l'évaluation en sont un exemple, d'autres courants cherchent à installer des méthodologies à vocation universelle. On mentionnera ici celle proposée dans le cadre du projet Esprit HUFIT consacré à la mise au point d'outils facilitant l'intégration des facteurs humains dans la conception des système d'information (Novara et al, 1987 a et b). Cette méthodologie comporte 5 étapes étalées sur une durée totale de 12 jours.

- la première étape consiste à élaborer un *profil des utilisateurs* prenant en compte leur expérience en informatique pour constituer des groupes expérimentaux.

- la seconde étape permet aux sujets de s'entraîner à l'utilisation du dispositif en réalisant des tâches *d'apprentissage* sans limite de temps mais en satisfaisant un critère de performance (nombre minimal d'erreurs). Au cours de cette phase, ils doivent remplir une "fiche d'utilisation" chaque fois qu'ils se servent du traitement de texte. Ces fiches facilitent l'estimation du temps nécessaire pour atteindre le critère d'apprentissage et comportent les informations suivantes:

    - fonction utilisée
    - type d'assistance préféré (manuel, tutoriel, aide contextualisée)
    - utilité de l'aide fournie
    - difficultés rencontrées

- au cours de la troisième étape, les sujets sont *testés individuellement*. Les tâches consistent à manipuler des fichiers, à créer de nouveaux textes à partir de textes existants et à utiliser de façon intensive les commandes de l'éditeur. Une fiche d'observation est établie pour chacun des sujets et pour chaque condition testée. Chaque fiche comporte les performances enregistrées (durée de réalisation, nombre d'erreurs, demandes d'aide), les commentaires des sujets au cours de la passation et les observations de l'expérimentateur.

- un *entretien post-expérimental* permet à l'expérimentateur de valider les stratégies d'apprentissage qu'il a pu identifier. Les sujets expriment leurs impressions sur le logiciel et celles-ci sont enregistrées ainsi que leurs questions.

- la dernière étape consiste à remplir un *questionnaire d'utilisabilité* dans lequel les sujets doivent donner une note de facilité d'usage, une note de facilité d'apprentissage et une note d'utilité de chaque caractéristique. Une note globale est fournie avec les suggestions d'amélioration

Les travaux conduits dans le cadre d'HUFIT suggèrent qu'une méthodologie d'évaluation satisfaisante suppose non seulement la disponibilité d'un arsenal de techniques mais surtout des

critères permettant de les mettre en oeuvre à bon escient. L'objectif est alors de mettre en relation les différentes phases du cycle de vie d'un produit et les outils de contrôle de qualité qui y sont les plus adaptés. En l'état actuel toutes les méthodologies proposées comportent des lacunes, et l'un des aspects critiques tient à la construction de *bancs d'essai de tâches* faisant apparaître les défauts de l'interface. Sur ce point, des efforts particuliers ont été conduits dans le cadre des **évaluations comparatives** présentées maintenant.

# CHAPITRE II.

# EVALUATIONS COMPARATIVES DE LOGICIELS VERTICAUX

La gamme de logiciels disponibles dans un domaine (traitement de texte, tableur, graphique...) est à la fois très variée et en constante évolution. Du fait de la rapidité du développement, il devient essentiel dans tous les secteurs professionnels de disposer d'une méthodologie comparative permettant de déterminer rapidement les qualités et les défauts d'un logiciel, de le situer par rapport à ses concurrents et d'établir les décisions d'achat sur des critères objectifs. Ce type d'évaluation pose des questions méthodologiques spécifiques du fait que les applications considérées présentent des caractéristiques très variées: elles exploitent des modèles conceptuels différents et chacune comporte des fonctionnalités particulières. Les évaluations comparatives des logiciels reposent sur des bancs d'essai combinant des tests empiriques et analytiques et dont la construction suppose le choix adéquat d'une population d'utilisateur et des tâches cernant de façon satisfaisante les qualités du produit.

> Les *bancs d'essai "orientés produit"* en mesurent les performances techniques (combien de temps faut-il pour réaliser les opérations ...?), identifient sa capacité fonctionnelle (que peut-il faire ?) et la qualité de l'assistance clientèle (jusqu'à quel point le vendeur supporte-t-il le produit?), Ce contrôle d'utilité établit l'adéquation du produit par rapport aux objectifs de l' utilisateur.
>
> Les *bancs d'essai de tâches* fondent méthodologiquement les évaluations en testant l'utilisabilité des IHM à travers leur facilité d'usage et l'adéquation de leur documentation.

Cette direction de travail est présentée en détail à travers deux exemples: l'un consacré aux éditeurs de textes, l'autre ayant une vocation plus générale et intégrant des contrôles d'utilité.

## 1. Comparaison de l'utilisabilité d'interfaces

### *1.1. Principe de l'évaluation*

La méthodologie d'évaluation proposée par Roberts et Moran (1983) compare des éditeurs de texte en faisant abstraction des caractéristiques spécifiques des produits (facilités locales, propriétés particulières...). Elle repose sur des tâches d'édition définissant les propriétés communes à l'ensemble des éditeurs et est développée en 2 temps: la première étape est la construction d'un banc d'essai de tâches, la seconde est la mise au point des mesures utilisées pour l'évaluation.

a) Construction d'un banc d'essai de tâche

Une taxonomie identifiant 212 tâches d'édition pouvant être assurées avec un éditeur de texte est élaborée. Les tâches sont décrites d'une façon indépendante de l'application considérée. Sur cette base, 32 tâches de base qui **doivent** être assurées par n'importe quel éditeur permettent de construire les 53 tâches-test composant le test de performance. Ce banc d'essai met en jeu des opérations sur 4 types de documents (un mémo, deux rapports de 2 pages, un chapitre de dix pages).

b) Système de mesure

La comparaison inter-éditeurs est assurée à partir de mesures sur 4 dimensions principales:

- la capacité fonctionnelle de l'éditeur
- la durée de réalisation des tâches d'édition
- la facilité d'apprentissage
- le temps de traitement des erreurs

Le schéma ci-dessous établit les relations entre les tâches et les mesures.

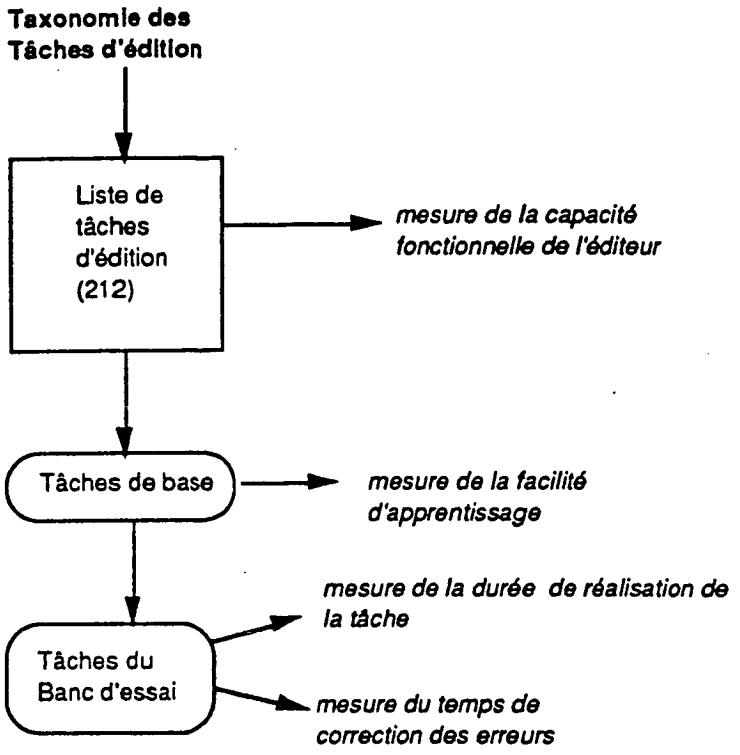



*Fig. 13: Principes de l'évaluation comparative de Roberts et Moran (1983)*

### *1.2. Mesure de la capacité fonctionnelle des éditeurs*

A partir de la liste des 212 fonctions d'édition possibles, des utilisateurs experts indiquent sur une échelle en 4 points la modalité de réalisation de chacune des tâches avec l'éditeur. L'échelle est la suivante:

- elle ne peut pas être réalisé
- elle peut l'être à la vitesse manuelle
- elle peut l'être maladroitement
- elle peut l'être efficacement

Les notes sont additionnées pour évaluer globalement chaque éditeur. Le résultat est une note qui exprime la capacité fonctionnelle de l'éditeur comme un pourcentage des fonctions que l'on pourrait attendre de celui-ci. Pour mettre en évidence les points forts et les faiblesses du logiciel, la note globale peut être décomposée en notes partielles correspondant à la classe de tâche dans la taxonomie.

### *1.3. Mesure de la facilité d'apprentissage*

L'apprentissage est mesuré expérimentalement en apprenant à un novice sans expérience de l'informatique comment réaliser les tâches de base avec l'éditeur. La séance d'apprentissage est réalisée en 5 cycles comportant deux étapes:

- apprentissage (2 à 5 heures)
- questionnaire permettant de savoir ce que le novice sait faire

L'apprentissage est mesuré en divisant le temps total de la session par le nombre total de tâches que le novice a appris selon le questionnaire (temps d'apprentissage par tâche). La note globale d'un éditeur est le temps moyen d'apprentissage pour les 4 novices.

### *1.4. Mesure des performances de l'utilisateur*

Un test expérimental consiste à évaluer les performances de 4 sujets experts sur chaque éditeur lors de la réalisation des 53 tâches d'édition du banc d'essai (il s'agit de tâches standards de saisie et de correction à partir de manuscrits). Les vitesses d'édition sont enregistrées à la montre. Pour chaque sujet, on obtient un temps moyen de performance de chacune des tâches. Le temps mis pour corriger les erreurs d'édition non triviales est enregistré et est soustrait du temps total pour établir le durée finale de réalisation d'une tâche. La durée globale de réalisation pour un éditeur est la moyenne des durées pour les 4 experts. Le test met en jeu une mesure analytique reposant sur le modèle KLM (cf chapitre III) prédisant la vitesse d'édition. Ces prédictions sont comparées ensuite aux résultats des expériences.

### *1.5. Mesure de la facilité de correction des erreurs*

Les auteurs indiquent que l'enregistrement des erreurs est difficile à réaliser: les erreurs graves se produisent rarement et le taux d'erreur varie considérablement d'un individu à l'autre. Le temps mis pour corriger les erreurs au cours des tâches du banc d'essai est cependant un bon indicateur de leur effet. La note d'erreur est ainsi la moyenne des temps d'erreur exprimée en pourcentage du temps sans erreur pour les 4 utilisateurs experts. Il est cependant nécessaire de définir une mesure plus fiable pour pouvoir différencier les éditeurs malgré les différences inter-individuelles.

### *2.6. Problèmes et limites*

Un certain nombre d'aspects n'ont pu être couverts par la méthodologie proposée, malgré les essais entrepris par les auteurs. Ils se sont intéressés par exemple:

- à la *sensibilité des éditeurs à l'erreur* en mettant les utilisateurs experts en situation d'utilisation sous stress.
- aux *possibilités d'erreurs désastreuses*, mesurées par une procédure d'analyse du langage de commande de l'éditeur
- à la *capacité d'affichage de l'éditeur*, évaluée par des tâches de lecture des écrans
- aux difficultés d'apprentissage et d'utilisation de *caractéristiques avancées*, étudiées via un questionnaire identifiant le savoir-faire lors de tâches d'édition complexes.

Toutes les mesures réalisées au cours de ces tentatives se sont cependant révélées trop grossières pour être fiables et pour différencier les applications. Les auteurs estiment cependant que cette méthodologie est facile à utiliser (ce qui est confirmé par d'autres travaux (Borenstein, 1985), facilité qui se paye par un certain nombre de limites. La durée de cette évaluation est élevée[1] mais la prédiction des performances effectives est satisfaisante, bien que le choix de tâches utilisées affecte incontestablement les résultats obtenus (par exemple, les mesures de facilité d'apprentissage sont très sensibles aux tâches effectuées).

La question de la représentativité des tâches se pose ici encore, car les bancs d'essai sont élaborés de façon indépendante de l'activité en situation naturelle: par exemple, ils ne reflètent pas les besoins apparaissant dans des situations d'exploitation particulières[2]. D'autre part, ils ne réalisent pas un véritable mélange de tâches, comme cela se produit fréquemment dans la réalité. Cet enchevêtrement crée des exigences que l'on ne parvient pas à cerner lors de l'évaluation et ne peut être abordé que par des études de terrain. Par ailleurs, la méthodologie met en jeu des experts, ce qui pose un vrai problème lorsqu'il s'agit d'évaluer l'interface d'un nouveau produit. L'idée intéressante proposée par les auteurs est dans ce cas d'utiliser des mesures analytiques pour

---

1 Un évaluateur expérimenté met environ une semaine pour évaluer un nouvel éditeur

2 Par exemple sur les 8 tâches d'organisation de paragraphes utilisées, une seule concerne la frappe de formules mathématiques.

assurer la comparaison: lorsqu'il existe déjà des logiciels assurant la classe de tâche, les performances d'utilisation du nouveau dispositif sont prédites en utilisant les modèles de tâche décrits ci-dessous (cf. chapitre III, KLM et Goms).

# 2. Comparaison de l'utilité des logiciels

## *2.1. Principe de l'évaluation*

La mesure de l'utilité d'un logiciel vise à déterminer dans quelle mesure il satisfait les besoins des utilisateurs. Cohill et al. (1988) proposent une méthodologie qui, bien qu'elle n'ait pas été validée expérimentalement, pourrait supporter des analyses systématiques. Elle n'est pas orientée vers l'utilisateur mais vers le produit et permet de l'évaluer en utilisant 5 catégories de critères. Trois de ces catégories (fonctionnalité, facilité d'usage et performance) peuvent être adaptées en fonction du domaine considéré (édition de texte, gestion de base de données, feuilles de calcul....). Ceci dépasse les limites des méthodologies usuelles dédiées à un type d'application (cf Roberts et Moran, 1983). Deux autres catégories (l'assistance clientèle fournie et la documentation) sont établies sur des critères génériques qui peuvent être appliqués à tous les types de logiciel.

Le principe de la méthode est le suivant: un logiciel est d'abord noté pour les critères d'une catégorie, puis, une analyse statistique est utilisée pour convertir les données en une note autorisant les comparaisons entre les produits. La pondération des données de base aux exigences spécifiques de l'environnement de travail repose sur l'analyse des tâches et de l'allocation des fonctions.

a) Dimensions de l'utilité d'un logiciel

Les attributs standards, à la base des comparaisons entre logiciels similaires (appartenant au même domaine) et mesurant l'utilité d'un produit sont les suivants:

- la capacité fonctionnelle: que peut faire le produit ?
- l'utilisabilité: facilité d'utilisation
- les performances du système: temps de réponse, vitesse,
- l'assistance technique: disponibilité, compétence...
- la documentation: adéquation, lisibilité...

Pour chaque attribut on identifie un ensemble de critères qui sont utilisés dans le processus de mesure numérique pour fournir des données cohérentes et fiables. Une procédure de pondération adapte l'information numérique aux besoins spécifiques des utilisateurs finaux.

b) Processus d'évaluation

Le processus d'évaluation comporte deux phases: une première étape vise à établir la liste des fonctions permettant de comparer les logiciels. Cette liste est établie en considérant tous les logiciels d'un domaine. Pour chacun d'eux, on dresse la liste des fonctions offertes à l'utilisateur. L'union de ces listes individuelles rend compte de l'ensemble des fonctions disponibles dans le doamine. A partir de cette liste source, une *liste fonctionnelle minimale* est élaborée: elle décrit les fonctions requises pour faire un travail utile et permet d'éliminer tous les logiciels qui ne satisfont pas ce critère. Le schéma ci-dessous résume cette procédure:

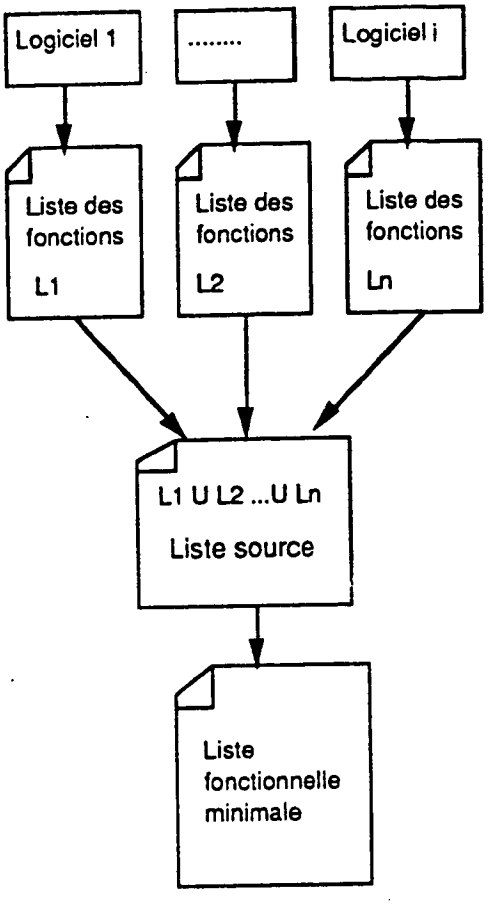



*Fig. 14: Construction de la liste fonctionnelle dans Cohill et al. (1988)*

Dans la seconde étape (mesure des applications), chaque logiciel est mesuré au regard des 5 catégories de critères. L'analyse des données fournit une note dans chaque catégorie de critères et permet de calculer la note moyenne du logiciel. Les critères opérationnels définissant les différents attributs sont présentés plus en détail dans ce qui suit.

### *2.2. Mesure de la capacité fonctionnelle d'un logiciel*

La mesure de la capacité fonctionnelle d'un logiciel exprime le nombre de tâches qu'il permet de réaliser. Les auteurs exploitent la méthodologie proposée par Roberts et Moran (1983) en établissant une liste de tâches réalisables avec un dispositif (elle comporte 400 items pour les bases de données). Cette liste est élaborée de façon itérative par des groupes d'experts. Une échelle en 3 points est développée selon que la tâche

- ne peut pas être réalisée
- peut être réalisée avec difficultés
- peut être réalisée facilement

Chaque logiciel est évalué au regard de sa capacité à réaliser chaque tâche de la liste. La note fonctionnelle d'un produit est la somme des notes obtenues à chaque item.

### *2.3. Mesure de la facilité d'utilisation*

Elle est définie comme la facilité (ou la difficulté) d'accés et/ d'implémentation des fonctions qui permettront à l'utilisateur final de réaliser les tâches d'application. Pour l'évaluation, KLM (cf chapitre III) est appliqué au même ensemble de tâches sur chacun des logiciels, ce qui facilite les comparaisons d'utilisabilité. Cette technique quantitative est objective, mais elle est jugée très coûteuse en temps. Cohill et al. (1988) utilisent alors une autre approche (BARS) qui associe à chaque objectif général ("facile à utiliser") une spécification plus précise ("simplicité"), des

critères opérationnels ("utilisation d'un vocabulaire ad hoc") et une échelle de mesure (ici le nombre de termes qui doivent être appris pour utiliser le système) qui aident l'évaluateur à rester cohérent.

### *2.4. Mesure des performances du système*

Un banc d'essai de performances techniques est développé à partir des tâches dérivées des fonctionnalités. La performance est mesurée à moindre coût par un opérateur utilisant une montre et le temps mis pour réaliser une tâche unitaire est une métrique concernant par exemple:

- le temps mis pour charger le logiciel
- le temps mis pour charger un fichier de donnée
- l'efficacité avec laquelle le logiciel manipule les données

Ces métriques peuvent être pondérées par l'utilisateur. Chaque mesure est répétée 5 fois et on en calcule la moyenne.

### *2.5. Evaluation de l'assistance technique*

Sur cette dimension, il s'agit d'établir une liste de critères concernant par exemple l'existence d'une assistance téléphonique (horaires ?), la disponibilité du personnel de maintenance (taille de l'équipe, expérience, affectation...), l'existence d'une lettre d'information (fréquence, longueur moyenne...), d'un groupe d'utilisateurs (indépendance, support par le vendeur ...) et les garanties en cas d'insatisfaction.

### *2.6. Evaluation de la documentation*

Celle-ci fait l'objet d'une analyse attentive car elle constitue souvent un point critique. Les aspects évalués concernent toutes les aides au travail destinées à informer l'utilisateur des accès aux fonctions, de leur utilisation et des facilités disponibles: manuel utilisateur, tutoriel, guide de référence rapide. Les critères mentionnés sont très nombreux: outre les caractéristiques matérielles de la documentation (taille et format), il s'agit d'évaluer:

- l'*organisation des documents* à travers l'existence d'une table des matières, d'index, de glossaire, et de vérifier que les chapitres, l'introduction et les résumés sont orientés par la tâche et non par les fonctions.
- la *typographie* et la *lisibilité* concernent la qualité de la frappe, la taille des fontes, le style, les règles de présentation (gras, italique...)
- la qualité de la *rédaction* qui détermine la facilité de lecture est appréhendée par la comptabilisation du nombre de syllabes dans les mots et du nombre de mots dans les phrases. D'autres indices concernent la fréquence d'usage des mots et la fréquence d'utilisation des voix active et passive. Ces analyses sont réalisées sur un échantillon de 5 passages de 150 mots tirés de la documentation qui subit un calcul automatique du niveau de lisibilité et du rapport actif/passif.
- les *graphiques* et les *illustrations* font aussi l'objet de l'analyse.

### *2.7. Notation des logiciels*

L'opération la plus critique se trouve dans la pondération de l'importance des fonctions. Elle doit être assurée en fonction des groupes d'utilisateurs: le poids d'une caractéristique varie pour les uns et les autres selon les tâches qu'ils réalisent. Chacune des dimensions d'évaluation ayant ses propres ensembles de critères et ses procédures de notation, une procédure statistique doit normaliser toutes les valeurs recueillies. La procédure compare les notes de performance non pondérées de chaque logiciel aux notes de performance non pondérées de l'ensemble des logiciels et établit une note sur une échelle de 0 à 4 points. La méthodologie permet un ajustement dynamique de telle sorte qu'à chaque évaluation d'un nouveau logiciel, l'échelle est réajustée pour qu'un produit soit toujours évalué par rapport à l'état du marché

Les évaluations comparatives sont à la charnière des approches empiriques et analytiques: elles mettent en jeu des tests de performance mais établissent aussi des comparaisons par rapport à des modèles de performances. Le second chapitre abordé maintenant présente de façon détaillée les principaux modèles utilisés dans les évaluations a priori de la qualité des IHM.

# Chapitre III.
# Approches analytiques de l'évaluation

EVALUATION A PRIORI DE LA QUALITÉ D'UNE IHM

Lorsqu'un diagnostic de qualité ergonomique doit être fourni sans que l'on puisse enregistrer de données relatives à l'utilisation, la situation d'évaluation est beaucoup plus contraignante que celles qui ont été présentées jusqu'à présent. Cette situation est rencontrée par exemple lorsque les contraintes de l'intervention ne permettent pas de construire une expérimentation bien contrôlée, de recueillir suffisamment de données, ou que les utilisateurs finaux ne sont pas disponibles. L'évaluation doit alors être réalisée en comparant l'IHM à un modèle de référence décrivant les propriétés de la "bonne interface". Dans l'idéal, un ensemble standard d'attributs décrivant l'objet évalué est mis en relation avec une échelle de mesure validée selon le schéma de principe suivant:

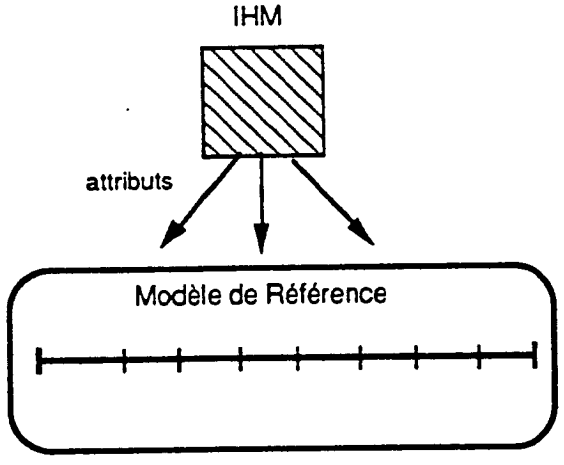



*Fig. 15: Schéma de principe de l'approche analytique*

Les difficultés d'évaluation tiennent ici à l'identification des dimensions pertinentes, à la construction des échelles de mesure et à l'intégration dans une *appréciation globale* de résultats caractérisant l'interface selon des points de vue très différents. Les modèles de référence utilisés pour poser un diagnostic a priori ne sont pas toujours formalisés ni même explicités: on peut les situer sur un continuum allant de l'*expertise ergonomique* à la *simulation de l'interaction.* Dans cette seconde partie on examinera successivement les modèles informels et les modèles formels des approches analytiques. L'architecture complète du chapitre est présentée dans l'arbre ci-dessous:

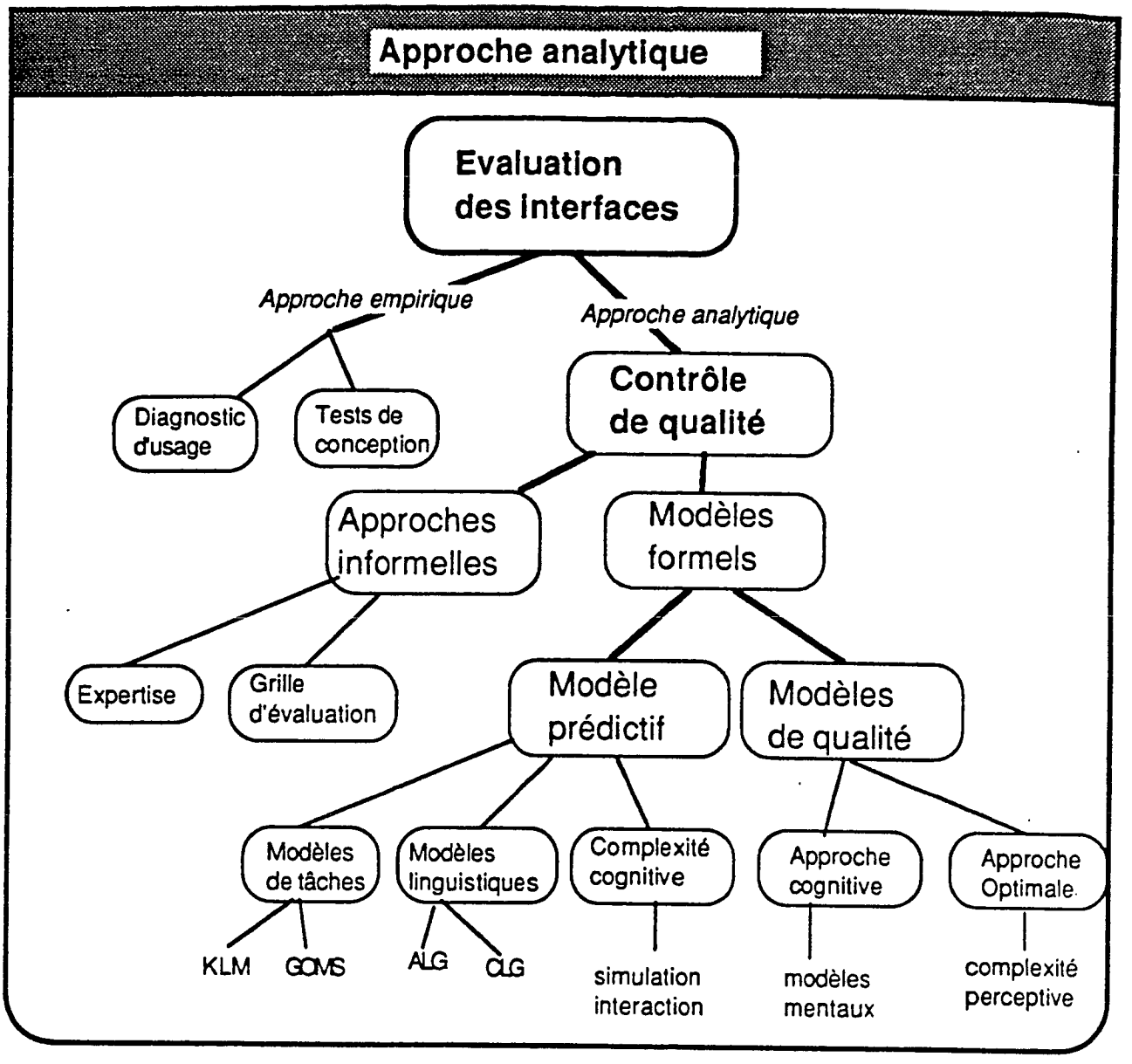



*Fig. 16: Structure du Chapitre III*

# I. Approches informelles

## 1. Expertise d'une interface

Les connaissances de l'expert constituent dans une certaine mesure un modèle de la "bonne interface": l'expert est censé savoir ce qu'il faut faire et ne pas faire pour améliorer la communication homme-machine, il est capable de reconnaître des défauts prototypiques fréquemment rencontrés, d'identifier les contraintes de l'interaction homme-machine à partir des exigences des tâches du domaine. Ces connaissances lui permettent une détection très rapide des problèmes potentiels; les hypothèses précoces sont validées ensuite par questionnement ou auto-utilisation. Une étude intéressante (Hammond & al., 1985) a contrasté les diagnostics fournis par des experts avec les évaluations obtenues par observations d'utilisateurs novices. En bref, les résultats font apparaître que les évaluations se recoupent mais qu'elles fournissent des informations à des niveaux très différents: le recueil auprès des utilisateurs identifie des difficultés conceptuelles et procédurales de bas niveau alors que les avis d'experts fournissent une vue plus générale du logiciel et conduisent à des hypothèses sur les sources potentielles de dysfonctionnement.

Bien que les évaluations par experts soient efficaces, le modèle de référence qu'ils utilisent repose sur un *savoir individuel* acquis par la pratique professionnelle et n'est pratiquement *jamais explicité*. Or, pour reprendre les exigences énoncées par Cohill & al. (1988), "l'identification des défauts et des qualités d'un logiciel doit pouvoir être réalisée sans que l'évaluation relève d'une opinion ou qu'elle mette en jeu des techniques d'évaluation personnelles. Les résultats d'une évaluation doivent de plus être présentés sous une forme standard, indépendante du produit et ils doivent mettre en évidence les aspects importants sans se disperser dans des détails".

Les jugements d'un expert sont biaisés par définition: en l'occurrence les évaluations d' IHM en ergonomie de l'informatique ne considèrent qu'un des multiples aspects de l'interface. En fait, une véritable expertise d'interface devrait faire intervenir d'autres spécialistes: psychologues cogniticiens, graphistes, rédacteurs de documentation technique. De plus, le savoir expert étant personnel, on peut se demander dans quelle mesure un biais de focalisation ne peut pas être introduit, autrement dit dans quelle mesure les problèmes les plus importants sont-ils identifiés? Dans l'étude mentionnée ci-dessus (Hammond & al., 1985) les évaluations d'une même interface par différents experts ont pu être comparées et l'on a montré que chacun des experts a tendance à se focaliser sur un thème récurrent. Par exemple, les uns mettent l'accent sur la nature des exemples dans la documentation, les autres sur l'utilisation du clavier ou sur les incohérences entre les exercices d'apprentissage. Il en résulte que c'est uniquement à partir d'une synthèse de leur rapport qu'il est possible d'avoir une évaluation complète de l'interface.

## 2. Grille d'évaluation ergonomique

### *2.1. Principe*

Le principe de l'outil consiste à fournir à l'évaluateur une liste aussi complète que possible des *propriétés d'une interface ergonomique*. L'évaluateur procède ensuite de façon systématique en notant sur une échelle de mesure (comportant généralement trois ou cinq points) chacune des propriétés de la liste. Par exemple, dans la grille d'analyse utilisée par Smith & al. (1984) chaque règle d'ergonomie proposée dans la littérature constitue une entrée de la grille d'évaluation. Cette transformation opérationnelle des recommandations des guides en une liste de propriétés requises génère théoriquement un modèle d'une interface ergonomique: un dispositif idéal aurait le score maximum pour chacun des items. Cependant la démarche comporte plusieurs limitations: outre les problèmes méthodologiques classiques associés aux échelles de mesure, la validité de la grille d'analyse dépend de trois aspects principaux:

- l'adéquation fonctionnelle des items de la liste pour le logiciel considéré
- l'énoncé des aspects qui doivent être analysés
- la notation des propriétés identifiées

## *2.2. Limites des grilles d'analyse*

### 2.2.1. Problèmes d'adéquation fonctionnelle

Les connaissances ergonomiques sont inégalement réparties: certains aspects de l'interaction homme-machine sont mieux connus que d'autres et l'évaluation est alors biaisée par l'état de l'art en matière ergonomique. L'analyse ne peut couvrir tous les aspects du logiciel considéré et certaines lacunes concernent des propriétés importantes. L'enrichissement d'une grille d'analyse pour tenir compte des fonctionnalités particulières d'une application n'est une solution réaliste que dans la mesure où l'on sait identifier les propriétés pertinentes , ce qui doit être réalisé sans que l'on puisse s'appuyer sur le savoir ergonomique.

D'autre part, le modèle de l'interface adéquate proposé dans la grille est "universel" en ce sens que la liste des propriétés est établie en extension, indépendamment des dispositifs techniques et des domaines d'application. La diversité des systèmes homme-machine rend caduque un nombre considérable d'items lors de l'évaluation d'un logiciel donné. C'est typiquement le cas d'attributs de codage couleur pour un moniteur monochrome. Une solution possible est de pondérer les différents items et de faire dépendre la notation des caractéristiques spécifiques de l'objet évalué. Autrement dit, des grilles d'évaluation spécifiques devraient être élaborées en fonction d'une taxonomie des IHM. Une autre idée mis en oeuvre par Tytyk (1988) pour l'ergonomie des postes de travail est d'appliquer la règle des 20-80 qui veut que 20% des éléments d'un ensemble soient responsables de 80% des propriétés du tout. Appliquée à l'ergonomie des interfaces, il s'agirait de hiérarchiser les critères selon leur importance et de réduire la liste sans limiter son pouvoir de diagnostic.

### 2.2.2. Evaluation subjective

Une autre limitation des grilles d'évaluation tient à l'interprétation qui doit être réalisée. Les items proposés dans la grille de Smith & al. (1984) induisent des réponses approximatives, très circonstancielles ou dépendant d'une appréciation individuelle. Par exemple, voici quelques aspects sur lesquels l'évaluateur doit se prononcer à propos de la *présentation des données:*

- cohérence du format
- adéquation de la taille des fenêtres
- organisation logique des données

Il n'y a pas de critères opérationnels définissant la "cohérence du format", l'"adéquation de la taille" ou la notion d'"organisation logique". De ce fait, la fidélité de l'évaluation est sujette à caution. Il manque à cet outil d'illustrer concrètement par des exemples ou des contre-exemples prototypiques la sémantique exacte de ces propriétés.

### 2.2.3. Pondération

Lors de l'utilisation de la grille, l'évaluateur doit juger le degré d'accord de chaque item avec l'IHM considérée. Plusieurs modalités peuvent être envisagées pour enregistrer les jugements. Des réponses binaires du type OUI-NON peuvent sembler suffisantes mais souvent une propriété n'est pas définie par sa seule présence/absence mais doit être située sur une dimension continue ou selon une gradation. Dans la grille de Smith & al. (1984), une échelle de mesure en 10 points permet à l'évaluateur de moduler son jugement. L'évaluation subjective prend une place importante par l'interprétation et la pondération de l'importance de chaque item, et le résultat du contrôle devient très sensible au niveau d'expertise de l'évaluateur.

Les jugements exprimés à travers des notes ne permettent pas de saisir la nature exacte des problèmes rencontrés durant la phase d'évaluation et la possibilité de décrire par des exemples les difficultés rencontrées (incident critique ou autre) est alors une aide considérable car les commentaires ne concernent pas toujours des défauts mais aussi des questions que se pose l'utilisateur. Ce type d'outil devrait de plus comporter un moyen de vérifier la cohérence entre

les réponses, comme cela a été signalé à propos du travail de Roots & al. (1983) relatif au questionnaire.

Les deux techniques que nous venons de survoler ont pointé les premiers problèmes des modèles de référence de l'évaluation analytique. Ces modèles sont constitués d'heuristiques, ne sont pas vraiment contrôlés et surtout ne sont pas validés.

# II. Modèles formels

Les modèles formels utilisés dans les approches analytiques élaborent des représentations abstraites des objets évalués permettant de prédire les performances des utilisateurs. Les modèles de l'interaction homme-machine sont construits à différents niveaux d'abstraction: ils peuvent décrire des tâches de bas niveaux comme "KLM" présenté ci-dessous (Card & al., 1983) ou formaliser précisément la notion de "complexité cognitive pour l'utilisateur" (Kieras & al., 1985). D'autres modèles rendent compte de la structure de l'interface (Moran, 1981) ou des procédures d'utilisation d'un dispositif (Reisner, 1981). Pour organiser ces différents courants, on a distingué ci-dessous deux classes principales de modèles: ceux qui veulent prédire précisément les performances des utilisateurs et ceux qui cherchent à définir les critères de qualité de l'interface. L'organisation est la suivante:

La première partie présente les modèles prédictifs des performances de l'utilisateur. Ces modèles formalisent la connaissance de la conception mais sont encore actuellement très limités. Leur intérêt tient à ce qu'il devient possible de prédire jusqu'à un certain point les performances d'utilisation à partir des seules spécifications de conception, autrement dit, il est possible d'évaluer une IHM sans avoir d'information directe sur son utilisation.

La seconde partie présente les modélisations de la qualité de l'interface. Dans le cadre de cette approche, il ne s'agit pas tant de décrire ou de prédire de façon précise l'utilisation d'un dispositif mais d'en identifier des propriétés formelles qui auront un effet sur les performances de leurs utilisateurs. Ces modèles visent à établir une correspondance entre propriétés et difficultés d'utilisation et doivent ensuite évaluer empiriquement la pertinence des analyses. Cette partie comporte deux sections opposant les approches cognitives de la qualité aux approches optimales.

# II.1. Modèles prédictifs des performances de l'utilisateur

## 1. Modèles de tâches

Les modèles de performance prédisent des durées d'exécution ou l'occurrence d'erreurs et ont été développés initialement à partir d'analyse de tâches interactives élémentaires sur des éditeurs de texte. Ils sont établis sur des mesures quantitatives: comptabilisation du nombre de frappes requises pour réaliser une tâche, vitesse de frappe, temps de réponse du système et temps de préparation mentale de l'action. Le principe général est de décomposer des tâches complexes en unités élémentaires dont on calcule le temps de réalisation. Il est ensuite possible d'inférer la durée d'exécution d'une tâche donnée en prenant en compte les contraintes imposées par un dispositif.

La valeur prédictive de ces modèles dépend bien sûr de l'exactitude avec laquelle l'estimation des durées des opérateurs élémentaires est établie et de la validité des hypothèses simplificatrices qui les sous-tendent. La validation des prédictions est alors une étape importante. Plusieurs modèles ont été proposés mais on mentionnera ici uniquement ceux qui ont été utilisés pour évaluer des IHM[1].

### *1.1. Keystroke- Level Model ("KLM")*

Le modèle "KLM" (Card & al., 1983) est un représentant de la classe de modèle "GOMS" (cf § 1.2.) qui situe ses unités d'analyse au niveau des actions physiques réalisées sur le clavier. Il prédit le temps d'exécution d'une tâche routinière par un utilisateur expérimenté ne commettant pas d'erreur. Dans ce modèle la prédiction du temps d'exécution d'une tâche complexe est réalisée par simple addition de la durée de réalisation de tâches élémentaires qui la composent, selon la formule;

$$T_{Tâche} = \Sigma\, T_{unités\ tâche}$$

où T représente la durée de la tâche.

#### 1.1.1. Décomposition des tâches complexes et calcul des durées

Une tâche complexe d'édition (création et correction d'un document par exemple) est décomposée en petites unités (sous-tâches) constituant des atomes élémentaires. Une unité peut correspondre à une commande d'un éditeur ou à une courte séquence structurée (type "couper-coller"). Pour l'essentiel, on dira ici qu'une unité de tâche est une sous-tâche pour laquelle le sujet dispose d'une méthode de mise en oeuvre et que l'on peut décrire de façon laconique (Embley & al., 1981). Le temps de réalisation d'une tâche élémentaire dépend du temps que met un utilisateur à acquérir une représentation mentale de la tâche et à l'exécuter, soit:

$$T_{unités\ tâche} = T_{acquisition} + T_{exécution}$$

Le temps d'acquisition est une évaluation du temps de préparation mentale nécessaire à l'action. Ce temps est supposé constant et est introduit lors de la description de la tâche à chaque fois qu'une décision doit être prise. Des heuristiques indiquent comment utiliser cet opérateur dans l'équation. Ces règles sont basées sur des hypothèses relatives au "chunking", (ie à la construction d'unités conceptuelles de plus en plus abstraites au cours de l'apprentissage). En bref les règles indiquent qu'une opération mentale est mise en oeuvre uniquement lorsqu'elle ne peut être associée à l'opération précédente.

---

[1] Le modèle proposé par Embley et al. (1978) n'est pas décrit dans cette présentation. Il est peu utilisé pour des raisons qui sont discutées dans Reisner (1984).

Le temps d'exécution global peut être décomposé par les temps d'exécution de 6 opérateurs de base[1] (pointage, rapatriement de la main, frappe de touche...), soit:

$$T_{exécution} = T_K + T_P + T_H + T_D + T_M + T_R$$

On ne rentrera pas plus avant dans les détails du modèle. Le lecteur intéressé pourra consulter Card & al (1983) et Embley & al. (1981). Le point important concerne la validation des prédictions établies à partir du modèle.

### 1.1.2. Validation du modèle

La validation a été réalisée à partir de mesures empiriques[2]: une analyse minutieuse des procédures d'utilisation d'un dispositif permet de calculer la durée de chacun des opérateurs (excepté pour les opérations mentales). La réalisation de chacune des tâches est ensuite simulée dans le cadre du modèle et les prédictions sont comparées aux résultats expérimentaux. Le modèle prédit la durée d'exécution des tâches avec une précision satisfaisante (les erreurs représentent environ 20% du temps moyen prédit). Pour la plupart des tâches, la performance enregistrée est proche des prédictions, mais dans certains cas, des écarts importants sont enregistrés. En intégrant à l'analyse une estimation du temps d'acquisition (préparation mentale de l'action), la précision est améliorée.

### 1.1.3. Problèmes et limites

KLM repose sur des hypothèses simplificatrices qui limitent son intérêt pour prédire les performances en situation naturelle. Par exemple, la prédiction concerne une performance sans erreur, or on sait que 5 à 30% du temps total d'édition peut être attribué aux erreurs et à la durée de leur récupération (Embley et al., 1981); on peut alors s'attendre à des écarts sensibles entre les prédictions et les mesures empiriques en situation naturelle (d'autant plus que le temps de réponse du système n'est pas pris en compte lors de l'analyse de la performance). L'évaluation d'une interface ne peut faire l'économie d'une modélisation des erreurs. D'autres limites tiennent à ce que l'opérateur de "préparation mentale de l'action" a une durée constante, alors que dans la réalité, l'individu peut se trouver en situation de résolution de problème.

L'utilisation de KLM suppose que soient définies:

- la tâche
- le langage de commande d'un système
- les capacités motrices de l'individu
- le temps de réponse du système
- une méthode de réalisation des tâches

La modélisation des tâches avec "KLM" est très longue (Cohill et al., 1988), ce qui la rend inappropriée selon le contexte d'évaluation. Par exemple, il est difficile de s'en servir lors des tests de conception compte tenu des exigences temporelles du développement (Novara et al.,

---

[1] Les 6 opérateurs sont les suivants.

- Keystroking = frappe de touche
- Pointing = désignation
- Homing = rapatriement de la main
- Drawing = dessin
- Mental = activité cognitive
- Response = réponse du système

[2] 12 sujets devaient réaliser 4 tâches d'édition selon 10 versions différentes en utilisant 3 éditeurs. Après une séance d'entrainement, les sujets devaient réaliser des tâches d'édition a partir d'un document manuscrit.

1987b). L'établissement d'une prédiction nécessite de définir a priori les méthodes qui vont être utilisées pour réaliser la tâche. Or, il n'est pas toujours facile de déterminer quelle méthode sera utilisée par un individu, celle-ci pouvant être sous-optimale (c'est d'ailleurs un des problèmes mentionné par les chercheurs qui ont utilisé KLM, voir Roberts et Moran, 1983). Enfin, les travaux de Kieras et Polson (1985) relatifs à la complexité cognitive (cf. Section 3) montrent que celle-ci dépend davantage de la "charge mentale" (ie du contenu de la mémoire d'un individu au cours d'une tâche) que du nombre de frappes de touches, ce qui réajuste la portée du modèle.

On peut retenir que si le modèle "KLM" est insuffisant pour constituer un outil d'évaluation à part entière, il constitue un élément important des méthodologies d'évaluation ( Cf. Roberts et Moran, 1983; Cohill et al., 1988) parce qu'il fournit une valeur numérique simple qui représente l'effort requis par un "expert standard" pour accomplir un ensemble de tâches pré-définies.

## *1.2. Goals, Operators, Methods, Selection rules (GOMS)*

"GOMS" (Card et al., 1983) décrit les performances de l'utilisateur d'un système donné en termes des buts poursuivis, d'opérateurs (actions élémentaires mises en jeu pour satisfaire le but poursuivi), de méthodes (définissant les procédures permettant d'atteindre le but) et de règles de sélection des méthodes. "GOMS" ne se contente pas de décrire la structure de la tâche mais tente de décrire le comment, i.e. de modéliser la façon dont l'individu s'y prend pour réaliser ses tâches. Il s'agit, comme pour "KLM", d'analyser la mise en jeu de routines cognitives ("cognitive skills") i.e. de connaissances fortement structurées par une pratique et mises en jeu dans des situations usuelles.

### 1.2.1. Le modèle

GOMS utilise une approche hiérarchique descendante décomposant une tâche en un arbre de buts et de sous-buts. Les prédictions de GOMS sont assurées en affectant une valeur aux opérateurs (actions élémentaires dont l'exécution provoque un changement d'état). Ils peuvent avoir une opérande d'entrée mais sont précisément définis par leur effet (sortie) et leur durée spécifique (temps d'exécution). Un exemple simple est l'opérateur de frappe de texte dont l'entrée est le texte à produire, la sortie est la séquence de touches frappées au clavier et dont la durée approximative est une fonction linéaire du nombre de caractères du texte. Soit graphiquement:

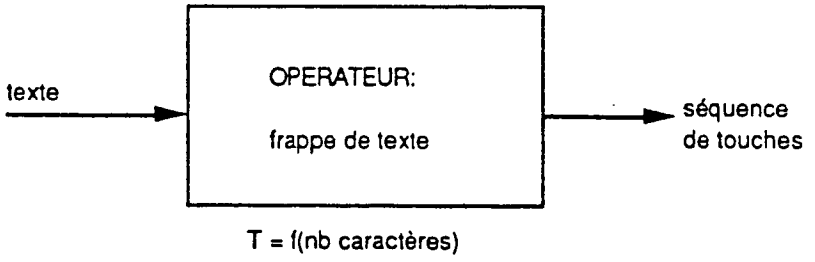



*Fig. 17: Exemple d'opérateur dans GOMS*

Un aspect intéressant tient à ce qu'on peut moduler la finesse de l'analyse en altérant la complexité des opérateurs: il est possible de décrire une tâche à différents niveaux d'abstraction, jusqu'au niveau le plus détaillé représenté par "KLM".

### 1.2.2. Validation du modèle

Les prédictions du modèle concernent la durée de réalisation des tâches, le choix des méthodes et le choix des opérateurs. La validation est assurée en deux temps selon le principe présenté pour KLM: description "en GOMS" de l'utilisation d'un dispositif, prédictions à partir du modèle et comparaison à des performances empiriques. Par exemple, les règles de sélection d'une méthode sont identifiées par observation, elles sont utilisées dans le modèle pour prédire la performance

dans un contexte précis et la prédiction est validée par une expérience plaçant l'individu dans ce contexte. Un ensemble d'études a été réalisé pour décrire les critères de choix d'une méthode de tâche donnée, pour rendre compte des séquences d'événements au cours de la réalisation d'une tâche et pour évaluer la précision des descriptions en fonction du "grain" de l'analyse. Leurs résultats mettent en évidence, entre autres, les points suivants:

-le choix des *méthodes* de réalisation des tâches peut être prédit dans 80 à 90% des situations d'utilisation.
- le choix des *opérateurs* peut être prédit dans 80 à 100% des situations selon le grain de l'analyse. La prédiction de la *séquence d'opérateurs* utilisés tombe à 50% lorsqu'on se situe au niveau le plus détaillé ("KLM").

#### 1.2.3. Problèmes et limites

Les auteurs concluent qu'il est possible de modéliser le comportement de l'utilisateur d'un éditeur de texte à partir d'une théorie cognitive comportant un petit nombre d'éléments du type "GOMS". Le point important est de déterminer le niveau de détail utilisé dans le modèle, car la qualité de certaines prédictions dépend du grain d'analyse (la focale). Par exemple, tant que le modèle décrit des unités de tâches macroscopiques, la précision reste satisfaisante, mais lorsque les unités sont plus précises, si la prédiction du temps d'exécution reste correcte, celle qui concerne les séquences d'opérateurs se dégrade.

Au total, "GOMS" fournit les éléments essentiels d'une modélisation structurée des tâches d'un domaine pouvant être exploités de façons très diverses (voir ci-dessous par l'exemple l'utilisation qu'en font Kieras et Polson, 1985).

## 2. Modèles linguistiques de l'interface

Dans les modèles précédents, l'accent est mis sur la structure des tâches. Les contraintes introduites par le dispositif restent implicites et n'apparaissent qu'à travers les méthodes pouvant être mises en oeuvre pour réaliser la tâche. L'approche linguistique de l'interface essaie de rendre explicite la structure de cet objet complexe sous la forme d'une grammaire. Les deux modèles principaux développés dans cette perspective sont le modèle Action Language Grammar de Reisner (1981) Le Command Language Grammar de Moran (1981).

### *2.1. Action Langage Grammar ("ALG")*

#### 2.1.1. Le modèle

L'objectif initial de Reisner (1981) est de construire un modèle des actions mises en jeu par l'utilisateur d'un terminal pour en faire un outil de conception de système ayant une valeur prédictive. Cet outil doit rendre possible:

- la comparaison des alternatives de conception du point de vue de leur facilité d'utilisation
- l'identification des choix de conception qui peuvent conduire les utilisateurs à commettre des erreurs.

Reisner considère que les actions de l'utilisateur définissent un langage dont on peut rendre compte dans une grammaire. "ALG" est une adaptation du formalisme BNF dans laquelle un ensemble de règles de production décrit les procédures d'utilisation d'un dispositif. En bref, la grammaire décompose récursivement les buts de l'utilisateur: le symbole initial de la dérivation est constitué d'un but général (eg **faire un dessin**), les symboles non terminaux (les phrases du langage) sont les **procédures d'utilisation**, les symboles terminaux (les mots du langage) sont les **actions élémentaires**. Les règles de la grammaire établissent la correspondance entre les buts et les opérations à mettre en oeuvre (du type: pour"...." faire"..."). Par exemple quelques règles de ré-écriture sont présentées ci-dessous pour le but initial "faire un dessin" (le symbole "->" se lit "se décompose en..")

| | | |
|---|---|---|
| Faire un dessin | -> | faire forme colorée<br>**OU**<br>faire un dessin **ET** faire forme colorée |
| Forme colorée | -> | choisir une couleur **ET** choisir une forme<br>**OU** choisir une forme **ET** choisir une couleur |
| Choisir une couleur | -> | CURSEUR DANS BLEU<br>**OU**<br>CURSEUR DANS ROUGE<br>**OU** ... |

### 2.1.2. Métriques utilisées

Dans une première étude, ce formalisme est utilisé pour comparer deux éditeurs graphiques ayant deux interfaces différentes (il s'agit de deux versions successives d'un même système). L'analyse met en évidence des incohérences entre procédures et définit surtout trois métriques relatives à la taille du lexique de commande, à la longueur des procédures et à leur cohérence qui sont utilisées comme indices de complexité du langage. Plus précisément:

- le nombre de symboles terminaux rend compte du nombre d'actions différentes pour atteindre un but. Pour Reisner la taille du lexique d'action à mettre en oeuvre est un indice de "complexité du langage".

- la longueur des séquences pour une tâche donnée est un indice de la "simplicité des procédures"

- le nombre de règles non nécessaires (ie deux règles sont requises là où une seule pourrait suffire) et le nombre de règles pour les séquences terminales similaires (enregistrement et sélection d'option dans un menu) fournissent un indice de "cohérence de structure"

### 2.1.3. Validation du modèle

Des prédictions (de difficulté d'utilisation, d'occurrence d'erreurs spécifiques) établies à partir du modèle ont été validées par des expérimentations préliminaires. La méthode consiste à comparer les prédictions du modèle avec des résultats expérimentaux mettant en évidence les difficultés d'utilisation. Divers problèmes méthodologiques ayant empêché une validation complète du modèle, Reisner (1984) a proposé une autre modélisation et réalisé une seconde étude expérimentale conduite cette fois avec un éditeur de texte. La grammaire utilisée s'efforce de caractériser la charge mentale liée à l'utilisation d'un langage de commande: elle combine des symboles décrivant les actions physiques avec des symboles décrivant des opérations cognitives (calcul mental, recherche en mémoire...). La méthode utilisée est un peu plus complexe que la précédente: elle commence avec une description grammaticale mais les "phrases" sont converties en équation de temps et d'erreurs sur lesquelles des calculs de prédictions sont ensuite effectués. La validation de ce modèle n'est pas encore achevée.

### 2.1.4. Problèmes et limites

Les critères proposés par Reisner (1981) pour établir la complexité d'un langage ne sont pas suffisants: il n'est pas possible de limiter la complexité aux seules données quantitatives considérées. En effet, si la description proposée est intéressante pour comparer des versions successives d'une IHM, elle présente des limites en tant qu'**outil de spécification et d'évaluation** d'un langage de commande. Sur ce dernier point, les critères de complexité et de simplicité sont trop réducteurs: la facilité d'utilisation n'est pas une simple fonction du nombre de règles et de la longueur des règles: des tâches faciles à décrire peuvent être difficiles à réaliser.

D'autre part, la description est trop indépendante de l'environnement matériel: il n'y a pas de référence aux moyens de contrôle-commande mis en oeuvre, ni des modalités de dialogue (menu, formulaire...). Enfin, l'analyse concerne la syntaxe d'entrée et ne dit rien de la syntaxe de sortie: il n'y a pas de relation avec les affichages (que se passe-t-il sur l'écran ?, où et comment sont affichés les objets ?). La dynamique du dialogue est ignorée (l'utilisateur est-il guidé ?, quels retours d'information sont fournis ?, que se passe-t-il en cas d'erreur ?...).

Le formalisme utilisé permet d'exprimer aisément la syntaxe d'un système: les grammaires de ré-écriture constituent la méthode de description conventionnelle des règles syntaxiques des langages de programmation. Mais, ces grammaires ne rendent pas compte de la sémantique. Or, dans le contexte de l'interaction homme-machine , la sémantique du système doit être reliée aux objectifs, aux buts des utilisateurs et au comportement du dispositif lui-même (Kieras et Polson, 1985). La représentation BNF n'est pas bien adaptée: elle conduit à décrire des séquences qui sont sans signification pour le dispositif et ne rend plus apparente la structure hiérarchique du dispositif (les modes ou les sous-états ne sont plus décrits).

## *2.2. Command Language Grammar ("CLG")*

Moran (1981) a proposé un modèle de l'interface qui, bien qu'il soit plutôt destiné à supporter la conception, permet potentiellement une évaluation prédictive de la facilité d'utilisation.

### 2.2.1. Le modèle

"CLG" utilise une approche hiérarchique descendante décomposant un système technique en différents niveaux d'abstraction. Le modèle comporte trois composants principaux qui peuvent eux -mêmes être décomposés. En voici une représentation graphique simplifiée:

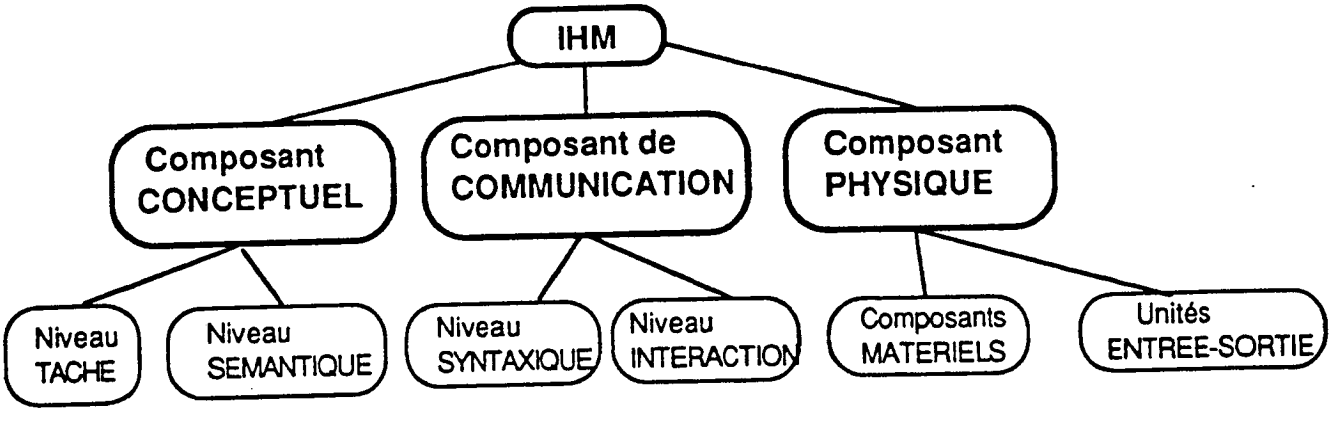



*Fig. 18: Modèle simplifié de l'interface dans "CLG"*

Une description complète de l'IHM en CLG est une série de modèles établie à chacun des niveaux en utilisant des règles reliant de façon cohérente chaque niveau. La grammaire CLG transforme la description des tâches en une description sémantique puis permet de passer de celle-ci à une description syntaxique, et ainsi de suite. Cette décomposition induit une conception descendante qui présente quelques inconvénients (voir à ce sujet Sharatt, 1987).

### 2.2.2. Utilisation de CLG pour l'évaluation d'interface

Selon Moran, "CLG" supporte l'évaluation car le modèle localise les décisions de conception qui sont critiques pour les difficultés d'utilisation ultérieures. Il propose des métriques d'évaluation de la cohérence de la conception suceptibles de prédire l'efficacité de l'interface par une estimation de vitesse d'exécution (en utilisant des formules du type "KLM") ou par le contrôle de la "charge mentale" de l'utilisateur ou encore par une estimation du temps d'apprentissage au niveau de chaque composant. Sharatt (1987) a exploité cette idée en cherchant à l'opérationnaliser. Il retient trois métriques utilisables pour évaluer la conception.

- une métrique de complexité reposant sur la structure des tâches et la longueur des méthodes syntaxiques
- une métrique d'optimalité analyse les composants de "CLG" en termes de leur structure et de leurs opérations
- une métrique d'erreur qui localise 5 sites potentiels de conception pouvant être à l'origine de difficultés des utilisateurs

  - la contrainte d'ordre d'expression des arguments dans les commandes à plusieurs paramètres (cohérence syntaxique)

  - l'utilisation de différents contextes pour une commande: l'entrée dans chaque contexte de commande induit la possibilité d'erreurs de mode.

  - l'utilisation de commandes génériques ("quit") pour assurer des transitions locales (retourner à des contextes précédents)

  - l'absence de feed-back (modification d'écran) lors de séquence de commandes à un argument

  - l'utilisation de commandes sans argument destinées à montrer à l'utilisateur le résultat de séquences précédentes

### 2.2.3. Problèmes et limites

Cette approche n'est pas directement exploitable pour évaluer un produit fini développé avec une autre méthodologie que "CLG": il faudrait traduire une IHM en "CLG" pour en faire l'analyse. On peut cependant en retenir l'idée de sites critiques de conception pour hiérarchiser les aspects de l'interface qui doivent être évalués en priorité. Une analyse systématique conduite dans cette perspective pourrait permettre d'identifier les points clés.

# 3. Modèles cognitifs de l'interaction

## *3.1. Prédiction de la complexité pour l'utilisateur*

Kieras et Polson (1985) veulent formaliser la notion de complexité du point de vue de l'utilisateur et en développer une analyse quantitative rigoureuse. L'hypothèse de base est que la complexité d'un dispositif relève d'un problème cognitif: elle dépend de la quantité, de la nature et de la structure des connaissances requises pour le manipuler efficacement. Selon les auteurs, les connaissances mises en jeu par un utilisateur lorsqu'il est confronté à l'utilisation d'un dispositif comportent deux composantes mentales principales. L'individu met en jeu:

- une représentation de la tâche à réaliser
- une représentation du dispositif

Dans cette perspective, la complexité pour l'utilisateur dépend alors

- de la complexité de la représentation de la tâche: elle détermine des exigences d'apprentissage, de mémorisation et a un impact sur les capacités de traitement du sujet
- du nombre de fonctions spécifiques du dispositif qui ne font pas partie de la représentation initiale de la tâche et de leur difficulté d'apprentissage
- de la facilité avec laquelle l'utilisateur peut acquérir la compréhension du fonctionnement

### 3.1.1. Le modèle

A partir de cette analyse, les auteurs développent un modèle de l'utilisateur et un modèle de l'interface en les décrivant de telle sorte qu'il devient possible de simuler l'interaction entre l'utilisateur et le dispositif. La logique des descriptions est en effet telle que:

> les sorties du modèle représentant l'utilisateur sont les entrées du modèle représentant le dispositif
> les sorties du modèle représentant le dispositif sont les entrées du modèle représentant l'utilisateur

Sans entrer dans des détails trop techniques, on dira ici que l'utilisateur est représenté comme un système de production (pour une présentation générale, voir Anderson, 1976)[1]. La représentation de la tâche y est assurée par une décomposition hiérarchique manipulant les entités décrites dans le modèle "GOMS" (buts, opérateurs, méthodes, règles de sélection). La représentation du dispositif est fournie dans un formalisme particulier (réseaux de transitions généralisés (GTN)) qui enrichit les diagrammes de transitions usuels en permettant des représentations hiérarchiques à différents niveaux d'abstraction. Elle est fournie sous une forme compatible avec les connaissances requises pour la manipuler.

La simulation de l'interaction est réalisée à partir de deux interprètes opérant sur chacune des représentations qui interagissent pour rendre compte des échanges d'information entre l'utilisateur et le dispositif. Le schéma ci-dessous décrit la structure du simulateur.

---

1 En bref, un système de production est un un programme qui comporte une mémoire de travail et des règles de production. La mémoire contient

- une représentation des buts courants
- des informations sur le statut des actions présentes et passées
- des représentations des entrées en provenance de l'environnement

Les règles de production sont des paires "condition -action" de la forme
SI (condition) Alors (action).
La condition est un énoncé relatif au contenu de la mémoire de travail (présence ou absence d'un but, présence ou absence d'une entrée venant de l'environnement). Si la condition est vraie, la production se déclenche et l'action est exécutée. L'action peut consister en une ou plusieurs actions élémentaires. Ce peut être entre autres

- insertion ou suppression de buts dans la mémoire
- insertion ou suppression d'opérations sur l'environnement

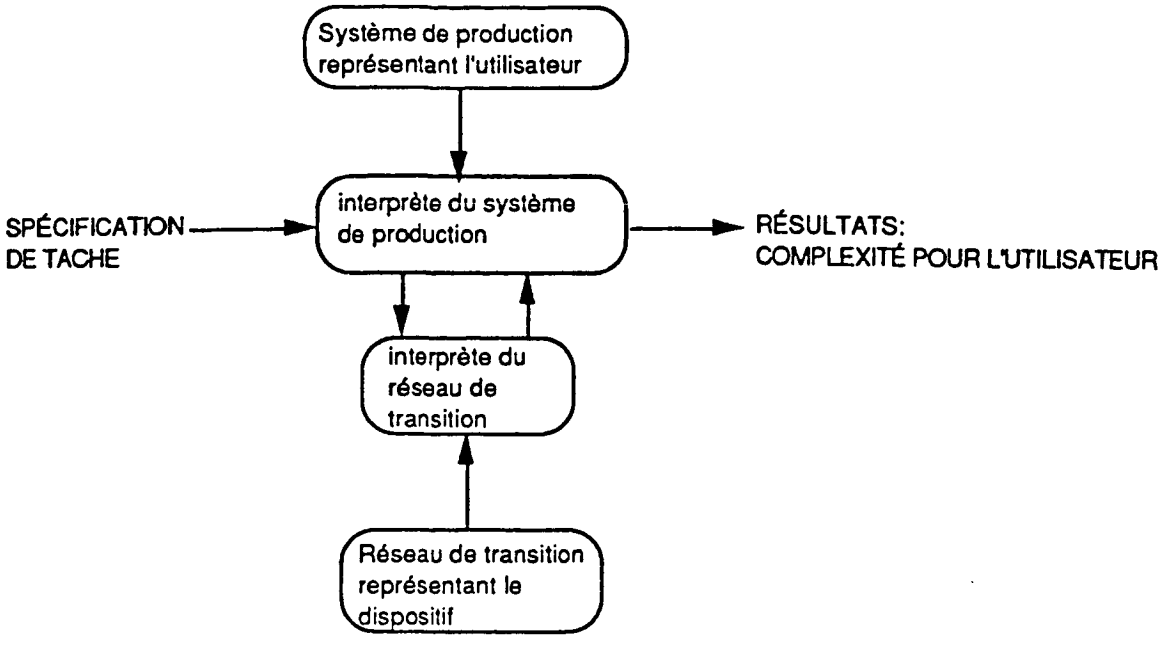



*Fig. 19: Simulateur de complexité cognitive (Kieras et Polson, 1985)*

La spécification des tâches devant être réalisées définit les entrées de l'interprète du système de production. Les sorties de la simulation définissent la complexité pour l'utilisateur, telles qu'elles ont été identifiées par l'interprète du système de production. Cette approche permet d'une part le développement de spécifications formelles théoriques des déterminants de la complexité pour l'utilisateur et d'autre part de tester la validité des hypothèses relatives à la tâche.

### 3.1.2. Mesures de complexité

Les indicateurs pertinents retenus par les auteurs pour caractériser la complexité sont entre autres les suivants:

- le nombre total de règles de production requises pour modéliser la tâche
- le nombre de production déclenchées
- le nombre de condition et d'action d'une production
- les piles de buts maintenues en mémoire de travail
- le nombre maximal de buts en mémoire pour réaliser une fonction donnée

### 3.1.3. Validation du modèle

Pour illustrer l'intérêt de cette représentation formelle pour la conception de systèmes complexes, les auteurs réalisent une analyse détaillée des compromis liés au choix de méthodes d'interaction. Ils comparent deux méthodes de suppression d'un mot dans un éditeur de texte, différenciées par leur niveau de généralité. Dans la première, l'entité qui doit être supprimée est définie comme un mot (méthode spécifique), dans l'autre comme une chaîne de caractère (méthode générale). Les analyses montrent que:

- les deux méthodes impliquent exactement le même nombre de touches
- elles comportent le même nombre maximum de buts en mémoire

La méthode générale impose cependant une "charge mentale" plus importante:

- elle nécessite d'apprendre et d'exécuter davantage de règles
- elle met en jeu davantage de cycles d'exécution

- elle implique de maintenir en mémoire davantage d'items, durant une période plus importante

Les auteurs en concluent que les différences pertinentes pour caractériser la complexité ne se situent pas au niveau de la frappe des touches mais au niveau de la charge en mémoire et de l'importance des connaissances procédurales requises pour assurer une tâche. L'analyse quantifie précisément les compromis qui doivent être traités: si la méthode générale détermine la plus grande complexité au regard d'une procédure spécifique, il faudrait pour la remplacer développer un ensemble de procédures de suppression opérant sur des entités spécifiques.

### 3.1.4. Problèmes et limites

Si dans le principe, la simulation de l'interaction à partir de représentations compatibles est une technique d'évaluation intéressante, la question de la pertinence psychologique de la modélisation devient essentielle. Autrement dit dans quelle mesure l'activité d'un sujet humain peut -elle être formalisée dans un système de production ? On peut par exemple être réservé sur l'interprétation de la complexité:

- une idée implicite est qu'il n'existe qu'une modélisation possible des tâches: faire du nombre de productions un indice de la complexité suppose que deux analystes modélisant la représentation d'une même tâche par un système de production produiront le même modèle. Or ceci est loin d'être acquis et en tout cas demande à être validé. L'utilisation d'un formalisme contraint l'expression d'un modèle mais pas le processus de modélisation lui-même: Sharatt, 1987 montre que des représentations très diverses de l'IHM peuvent être élaborées à partir du formalisme proposé dans "CLG".

- un autre aspect tient à la prise en compte de différences inter-individuelles. Si la complexité pour l'utilisateur dépend effectivement des représentations élaborées par l'individu, et donc, entre autres des connaissance de la tâche dont il dispose, les mesures devraient être sensibles à l'expérience du domaine, mais cette dimension n'est pas traitée.

Le développement de modèle de simulation utilisant des représentations formelles devrait à terme fournir une méthodologie puissante pour la conception de nouveaux dispositifs. Cette approche devrait permettre aux développeurs de simuler des prototypes abstraits, de spécifier les connaissances requises pour en assurer la commande avant d'engager la construction de prototypes réels. Ceci suppose la définition de mesures quantitatives précises permettant d'évaluer la complexité pour l'utilisateur de diverses alternatives et de préciser les compromis qui doivent être trouvés. Ce courant reste actif et on en trouvera les développements récents dans Kieras (1988) et Polson (1987)

## II.2. Modèles de la qualité de l'interface

La plupart des formalismes qui viennent d'être présentés décrivent l'élaboration de séquences d'action permettant de satisfaire des buts de haut niveau d'abstraction. Pour la plupart, ils rendent davantage compte de ce que doit faire le sujet avec le dispositif pour réaliser sa tâche que de la structure de l'objet technique utilisé. C'est justement à cet objet que s'intéressent les modèles de qualité de l'interface. Ils cherchent à identifier des propriétés mesurables caractérisant les exigences que doit satisfaire l'interface utilisable (cohérence, lisibilité...) et comme toutes les approches analytiques, ils doivent valider leur conclusions (ici la pertinence des propriétés identifiées) par des mesures des performances. On peut différencier deux points de vue principaux sur la qualité selon que l'accent est mis sur les *aspects cognitifs* ou sur des *caractéristiques optimales.*

Ces différences "philosophiques" sont bien illustrées par les travaux traitant de la présentation d'information et qui s'intéressent de façon exclusive à la complexité perceptive ou à

la complexité cognitive des affichages. Dans le premier cas, la sémantique de l'information affichée n'est pas prise en compte et l'évaluation est établie sur des critères quantitatifs (nombre d'alignements horizontaux, verticaux, densité, nombre d'items....). Dans le second cas, l'évaluation de la qualité de la présentation d'information est faite en prenant en compte les traitements effectués sur ces informations; des critères externes sont alors utilisés: adéquation à la tâche, regroupements fonctionnels...

Cette seconde partie présente tout d'abord l'approche cognitive de la qualité à partir de travaux analysant la compatibilité des modèles mentaux des utilisateurs avec la structure du dispositif utilisé avant de traiter des approches optimales surtout orientées vers la qualité des affichages.

# 1. Approche cognitive de la qualité

L'approche cognitive de la qualité repose sur le leit-motiv de la psychologie ergonomique: "l'efficacité d'un système homme-machine dépend de l'adéquation des outils à la logique du travail des opérateurs". En d'autres termes, à un niveau à peine moins général, "la qualité d'une IHM dépend de sa compatibilité avec les représentations mentales élaborées par ses utilisateurs". Cette assertion traduit l'idée que l'utilisateur d'un dispositif technique est confronté à un double problème: celui du transfert de ses connaissances du travail et celui d'un apprentissage du fonctionnement du système. L'interface idéale facilite la mise en jeu des connaissances spécifiques du domaine dont dispose l'utilisateur et lui permet d'y intégrer les contraintes d'utilisation du dispositif. Enoncée à ce niveau d'abstraction, le corollaire est évident: pour améliorer la qualité des IHM, il convient:

- d'analyser la structure des connaissances acquises par la pratique professionnelle et de décrire les modalités de leur mise en jeu (comment sont-elles organisées, indexées, quelles sont les procédures d'accès, les déterminants de l'oubli...)

- d'identifier les conditions de l'apprentissage et d'en connaître suffisamment les mécanismes pour pouvoir les contrôler.

Une première direction de travail engagée dans cette direction analyse la structure des *modèles mentaux* des utilisateurs à partir de techniques expérimentales classiques. La seconde perspective tente de formaliser la notion de cohérence de l'interface, condition de l'utilisabilité d'une interface jugée essentielle mais rarement définie.

## *1.1. Modèles mentaux et navigation dans les menus*

La question de la recherche d'information dans les menus a pris une importance grandissante du fait du développement des applications vidéotex grand public et de la multiplicité des fonctionnalités des ordinateurs. Le nombre de commandes et de fonctions qui sont proposées aux utilisateurs des gros systèmes informatiques devient impressionnant: par exemple, la version de l'éditeur EMACS du système UNIX étudié par Palmer et al. (1987) comporte 110 commandes et le système d'exploitation étudié par Tullis (1985) en a 700 (!). Le temps passé par les utilisateurs à rechercher dans ces arborescences inextricables les moyens de réaliser leur travail devient critique et un ensemble de travaux ont tenté d'apporter des solutions à ce problème.

Une première direction de travail, qui ne sera pas développée ici, facilite la navigation dans les arborescences complexes: on trouvera par exemple dans Aperley et Field (1985) une solution élégante de parcours dans les hiérarchies de menus[1]. Une autre perspective tente de mettre en évidence les déterminants d'une organisation facilitant l'accès aux informations recherchées. L'efficacité d'une organisation dépend évidemment de la façon dont les utilisateurs se

[1] Un menu présentant la liste des options précédemment choisies autorise la retraite sélective: l'utilisateur peut revenir à n'importe quel niveau de l'arbre.

représentent le problème, i.e. des clés utilisées lors de la recherche. Par exemple, si l'utilisateur recherche une commande dont il connait le nom, une organisation alphabétique des listes d'items est tout à fait satisfaisante. Par contre, s'il ne connaît pas la commande, ce qui est fréquent, il est probable que la recherche repose sur un critère fonctionnel, défini par rapport au but poursuivi. Mais, ce but est souvent formulé à un tel niveau d'abstraction qu'il n'existe pas une seule commande pour le réaliser et que l'utilisateur doit en fait construire une séquence de commandes. Dans cette perspective, l'organisation des menus doit être calquée sur la façon dont l'utilisateur se représente la tâche lorsqu'il utilise le dispositif. Les travaux conduits dans cette direction veulent donc identifier la représentation mentale du système élaborée par l'utilisateur. Par exemple, Tullis (1985) analyse les relations logiques que les individus établissent entre les éléments d'un menu pour organiser l'interface d'un système d'exploitation; Palmer et al. (1987) et McDonald et al. (1986) abordent cette question à propos des environnements d'assistance, les premiers voulant faciliter aux utilisateurs d'EMACS l'accès au système d'aide de l'éditeur en structurant les thèmes d'intérêt; les seconds s'intéressant au MAN d'UNIX.

#### 1.1.1. Un paradigme d'étude: l'adéquation au modèle mental de l'utilisateur

Tous les travaux qui ont été mentionnés exploitent le même paradigme expérimental dans lequel les sujets classent des cartes comportant le nom des commandes (plus éventuellement d'autres informations: description fonctionnelle laconique, mnémonique...) selon des critères de similarité. Une procédure statistique (classification hiérarchique) est utilisée pour identifier les regroupements de commandes dont on infère ensuite la façon dont les sujets se représentent l'interface (le modèle mental du dispositif). Le *niveau d'expérience des utilisateurs* est généralement un facteur contrôlé dont on attend qu'il détermine des différences de regroupement. Dans une étape ultérieure, les résultats des analyses hiérarchiques sont utilisés pour construire des menus. L'efficacité de ces structurations censées être plus proches des modèles mentaux des utilisateurs est alors évaluée en prenant en compte les différences liées à l'expérience (les menus construits pour les novices sont-ils plus facilement utilisés par les novices ?).

#### 1.1.2. Problèmes et limites

La validité de la méthode d'identification des "modèles mentaux" pose deux questions principales ( Palmer et al.1987). La première tient à ce que les résultats de la classification sont très sensibles à la technique utilisée: selon les algorithmes mis en oeuvre, on n'obtient pas les mêmes regroupements. La seconde est de nature écologique et porte sur la pertinence de la tâche expérimentale utilisée: dans quelle mesure le classement de commandes permet-il d'identifier les représentations mises en jeu sur une position de travail. La différence essentielle entre ces deux situations est que les groupements traduisent une organisation thématique alors que les représentations élaborées en situation de travail sont organisées par des objectifs de travail de haut niveau. On retrouve dans cette analyse une distinction classique en ergonomie qui oppose les *représentations cognitives* et les *représentations opératives.*

### *1.2. Prédiction des difficultés d'utilisation par l'étude des modèles mentaux*

#### 1.2.1. Hypothèse de travail

Selon l'hypothèse de Kieras et Polson (1985) la complexité d'un dispositif dépend de l'écart entre les connaissances requises pour l'utiliser et les modèles mentaux élaborés par ses utilisateurs: plus le modèle mental est proche du modèle "objectif" du système, plus le dispositif est facile à apprendre et à utiliser. La difficulté est alors d'identifier des métriques permettant de contrôler les écarts entre ces deux représentations.

Kellog et Breen (1983) se sont engagés dans cette voie en faisant l'hypothèse d'une relation entre l'expérience de l'utilisation et l'exactitude du modèle du système que construit l'utilisateur: l'expérience acquise au cours de l'utilisation aurait pour effet d'ajuster progressivement le modèle initial de l'utilisateur au modèle "réel" du système (il s'agit à peu de

chose près de celui du concepteur). Pour vérifier cette hypothèse, les auteurs élaborent une méthodologie pour identifier ces deux représentations et les représenter sous une forme comparable. Il s'agit:

- de spécifier un modèle du système (qui sert de modèle de référence)
- d'identifier les modèles du système que se construisent des utilisateurs en fonction de leur niveau d'expérience
- de développer des méthodes permettant d'établir le degré d'adéquation entre modèle mental et modèle réel

#### 1.2.2. Etude expérimentale des effets de l'expérience sur la structure des modèles mentaux

Le travail expérimental est réalisé sur un système de formatage de texte (gestion de paragraphes, indentation...). Un modèle de ce système est d'abord élaboré à partir d'une technique originale d'analyse de sa documentation. Il en résulte un diagramme de structure différenciant les composants selon une organisation hiérarchique qui est estimé être la représentation la plus claire et la plus objective. Le modèle du système est basé sur une mesure de la distance entre les paires de concept dans l'ensemble des commandes, calculées à partir du diagramme de structure.

Le modèle mental que se construisent les utilisateurs est dérivé d'une tâche de classification classique (cf. ci-dessus). Les sujets doivent appareiller 51 cartes comportant chacune une commande de formatage en constituant des piles. Ces groupements sont considérés être des jugements subjectifs de la structure du système et constituent le modèle mental.

En bref, l'analyse fait apparaître des groupements différents selon le niveau d'expérience des individus. Les experts sont plus proches du système que les novices: ces derniers font reposer leurs classifications sur des ressemblances de surface et exploitent des relations sémantiques entre les termes, ce qui organise la grande majorité des concepts d'une façon aberrante, alors que les experts classent selon des critères fonctionnels et établissent des groupements cohérents.

Selon les auteurs cette analyse permet de prédire les difficultés qu'auront les utilisateurs experts à partir des groupements fonctionnels inadéquats. Dans la mesure où ces groupements traduisent effectivement la représentation mentale du dispositif, des procédures qui ne sont pas valides risquent d'être utilisées. L'idée force est que les déviations des classifications par rapport au modèle du système suggèrent une restructuration des commandes pouvant améliorer l'utilisabilité. Les déviations des "modèles" des novices sont exploitées dans les guides d'apprentissage en focalisant l'entraînement sur les points identifiés. La vérification des prédictions du modèle est envisagée mais n'a pas été réalisée.

### *1.3. Cohérence conceptuelle de l'interface*

La cohérence de l'interface est une propriété dont le poids sur l'utilisabilité des dispositifs est unanimement reconnue par les chercheurs et par les concepteurs. On admet par exemple que la cohérence aide l'individu à identifier des structures générales et des règles de fonctionnement (ou d'utilisation) qui lui permettent ensuite de réaliser des inférences: il peut prédire l'effet de commandes qu'il ne connaît pas, reconnaître des entités sur la base de ressemblances et transférer à de nouveau dispositifs des procédures acquises dans d'autres contextes (Payne et Green, 1986; Carroll et Thomas, 1982). A ce titre, la cohérence est un déterminant majeur de la facilité d'usage et d'apprentissage.

Cependant, les exigences de cohérence sont d'autant plus difficiles à respecter que la notion n'est pas opérationalisée. Les acceptions du terme sont très variables: elles portent soit sur des aspects conceptuels (métaphore utilisée, modèle de l'utilisateur...) soit sur des aspects sémantiques (effet d'une même touche dans des contextes différents) ou syntaxiques (organisation des séquences d'actions, position des arguments dans les commandes...). Dans ce qui suit, on parlera de cohérence interne pour tous les aspects qui relève des propriétés

intrinsèques de l'interface (par exemple le respect d'un système de contraintes lors de la définition d'un langage de commande). On parlera de cohérence externe pour tous les aspects concernant une correspondance avec des propriétés d'objets existant en dehors de l'interface (par exemple la congruence de la structure d'un langage de commande avec la structure de la langue naturelle).

### 1.3.1. Types de cohérence d'une IHM

Kellog (1987) a cherché à opérationaliser la notion de cohérence et à en mesurer les effets sur les performances des utilisateurs. L'auteur différencie les types de cohérence à partir du modèle de l'interface proposé par Moran (1981) présenté ci-dessus. Chacun des 3 composants principales d'une IHM (composante conceptuelle, de communication et physique) est décomposable en deux niveaux. A chacun de ces niveaux , Kellog caractérise la façon dont se manifeste la cohérence interne et externe. Le tableau ci-dessous résume le point de vue.

| NIVEAU | | | COHÉRENCE INTERNE | ADÉQUATION EXTERNE |
|---|---|---|---|---|
| | CONCEPTUEL | niveau tâche | décomposition de tâches analogue | correspondance entre tâche utilisateur et tâche système |
| | | niveau sémantique | procédures sémantiques<br>complétude | correspondance avec la sémantique de la tâche<br>correspondance métaphorique |
| | COMMUNICATION | niveau syntaxique | principes d'organisation<br>régularité du langage | |
| | | niveau interaction | principes d'organisation du lexique de commande<br>position des arguments | congruence du lexique de commande avec la langue naturelle |
| | PHYSIQUE | présentation spatiale | Disposition spatiale<br>attributs graphiques des objets | |
| | | Dispositif | | |

*Tableau 2: Cohérence de l'interface et niveaux d'abstraction*

Sur la base de cette analyse, Kellog cherche à mettre évidence les effets de la cohérence conceptuelle en élaborant une expérience faisant varier les procédures sémantiques pour des tâches analogues (cohérence de l'accès aux fonctions).

### 1.3.2. Etude expérimentale de l'effet de la cohérence conceptuelle

#### a) Procédure expérimentale

L'expérience consiste à faire réaliser à un groupe de 5 sujets ayant une expérience très hétérogène de l'informatique, une série de tâches sur trois interfaces différentes. Les tâches consistent à créer et à rechercher des objets sur l'interface, et les procédures mises en oeuvre sont, soit identiques quel que soit le type d'objet sur lequel elles portent, soit variables en fonction de la tâche et du type d'objet. L'une des interfaces est incohérente et la cohérence des

deux autres est rendue plus ou moins évidente en associant (ou en dissociant) certaines fonctions dans un même menu. Les mesures réalisées veulent mettre en évidence les effets de la cohérence conceptuelle dans 4 domaines:

- effet sur les performances de l'utilisateur
- effet sur la capacité de rappel et d'inférence
- évaluation subjective du système
- adéquation et robustesse du modèle mental acquis à travers le travail avec le système

b) Résultats

Les résultats montrent tout d'abord que le temps de réalisation des tâches est beaucoup plus long avec les interfaces incohérentes. L'aspect original de l'entreprise tient à ce que certaines tâches proposées pour évaluer l'effet de la cohérence sur les connaissances de l'utilisateur nécessitent la mise en jeu de ces connaissances: elles testent la capacité de rappel et d'inférence de l'individu. Par exemple, le sujet doit rappeler certaines procédures utilisées dans les tâches qui lui ont été proposées, il doit indiquer comment il pourrait réaliser 5 tâches qui n'ont pas été proposées au cours de l'expérience et il doit exprimer pour chacune de ses réponses la confiance qu'il leur accorde (exactitude de la réponse, efficacité). Ces tâches montrent que les performances de rappel et d'inférence des utilisateurs des systèmes cohérents sont nettement meilleures que celles des autres sujets: ils ne font pratiquement pas d'erreur alors que les autres ont beaucoup de mal à rappeler quelque chose. De plus, ces derniers infèrent des procédures cohérentes, qui sont des erreurs compte tenu du fait qu'elles ne peuvent pas être mises en oeuvre avec le système incohérent.

Ainsi, l'incohérence empêche les opérateurs d'avoir une idée structurée des moyens de réaliser des tâches routinières. L'intérêt essentiel de ce travail pour l'évaluation d'interface est dans la distinction des types de cohérence et dans l'utilisation de tâches originales pour en mesurer les effets.

# 2. Approche optimale de la qualité de l'interface

## *2.1. Un modèle behavioriste du comportement de l'interface*

### 2.1.1. Quatre critères d'évaluation de l'utilisabilité d'une interface

L'approche proposée par Monk et Dix (1987) est sensiblement différente de la précédente. Ces auteurs estiment que la facilité d'utilisation potentielle d'une interface peut être évaluée dès les premiers stades du développement par des techniques vérifiant le respect de principes de conception. Ils définissent 4 principes de base que toute interface doit satisfaire:

- la *prédictibilité* est une propriété grâce à laquelle l'utilisateur est en mesure de déterminer ce qui doit être réalisé après chaque interruption: il ne doit pas se référer aux actions antérieures pour prédire l'effet des actions ultérieures.

- la *simplicité* concerne les règles d'usage et les procédures qui devront être acquises par l'utilisateur pour se servir du système. Ce critère met en évidence la "complexité inutile" de l'IHM.

- la *cohérence* concerne uniquement la sémantique des commandes: l'analyse porte sur l'effet d'une action de l'utilisateur dans différents contextes. Par exemple, une règle indique que la touche "backspace" doit avoir le même effet quel que soit le contexte dans lequel elle est utilisée.

- la *réversibilité* concerne la possibilité d'annuler des commandes engagées par l'utilisateur.

### 2.1.2. Le modèle "black box"

La technique proposée consiste à décrire le comportement d'un dispositif par des **règles "action-effet"**. Dans ce modèle behavioriste, le dispositif constitue une "boîte noire": les actions de l'utilisateur (presser une touche, déplacer la souris...) sont décrites en termes d'effets observables et on ne dit rien des modifications des états internes.

L'évaluation repose sur une analyse qui consiste en premier lieu à identifier des contextes d'affichage (i.e. des écrans, par exemple "écran d'accueil", "menu principal"...). Puis, on définit un lexique des actions de l'utilisateur qui organise la liste exhaustive des actions élémentaires sur les dispositifs d'entrée (A = {ensemble des caractères éditables}, <espace>, <tabulation>, <effacement>...). Des règles "action-effet" sont ensuite rédigées pour chacun des contextes dans une notation semi-formelle. Elles consistent à associer à chaque action pouvant être réalisée dans un contexte déterminé, l'effet qui est observé. Par exemple la règle R2 du contexte C1 décrit l'effet de la touche d'effacement de la façon suivante:

R2. <effacement>:: s'il y a un caractère à la gauche du curseur, il disparaît.

### 2.1.3. Validation des critères

Il n'y a pas à proprement parler de validation des critères: les auteurs rendent compte d'une application de leur modèle et des aménagements qu'il permet de réaliser.

La *simplicité* de l'IHM n'est pas précisément définie: le critère pris en compte paraît lié au nombre et à l'énoncé des règles disponibles dans un contexte donné L'évaluation de la cohérence opère sur les règles qui ont été spécifiées sur l'ensemble des contextes. Les auteurs ne mentionnent pas la technique qu'ils utilisent pour détecter les incohérences; on peut supposer d'une part qu'une simple analyse de la partie" effet" des règles d'un contexte donné permet de découvrir les "incohérences sémantiques" intra-contexte (ie la même action ne déclenche pas les mêmes effets) et que d'autre part des comparaisons transversales (inter-contextes) évaluent une "cohérence sémantique globale". Pour évaluer la réversibilité, la partie "effet" de chaque règle est analysée au regard du nombre d'actions ultérieures qui seraient requises pour annuler cet effet. Par exemple, l'absence de commande d'annulation contraint l'utilisateur à achever la tâche qu'il avait engagé pour pouvoir revenir au menu principal.

### 2.1.4. Problèmes et limites

Si l'approche est effectivement intéressante car elle oblige le concepteur à formuler et traiter les problèmes qui se posent à l'utilisateur, elle reste pour l'instant insuffisamment formalisée. Les points suivants méritent d'être discutés:

- La violation d'un principe de conception ne détermine pas nécessairement un problème pour l'utilisateur. L'impact doit être estimé à partir d'une connaissance des tâches et des utilisateurs: par exemple, la simplicité dépend en grande partie des connaissances préalables des utilisateurs et l'importance des incohérences dépend de la fréquence de la tâche. Par exemple, le fait de ne pas pouvoir prédire les effets d'une commande n'est pas déterminant si les actions en question sont improbables dans le contexte courant (frappe d'une touche <escape>). Un autre exemple est que le niveau de détail des attentes conditionne l'estimation de la prédictibilité de l'interface: par exemple, si l'utilisateur attend un effet général sans pouvoir le spécifier précisément (défilement d'un affichage sans précision du sens) alors, tout effet sera satisfaisant. Les critères utilisés sont donc "à géométrie variable" ce qui laisse supposer des problèmes de fidélité de l'évaluation.

- Pour garantir la simplicité de l'interface les auteurs utilisent des solutions qui rigidifient considérablement l'utilisation. Par exemple, dans le système évalué, lorsque l'opérateur commet une erreur lors de la frappe d'un nom de fichier, s'il utilise la touche <arrière> ou <flèche gauche> pour effectuer la correction en pressant la touche <retour> pour changer de contexte, il obtient le résultat suivant:

un message d'erreur est affiché puis l'écran est effacé et re-dessiné. Ces affichages sont liés à ce que les touches <arrière> et <flèche gauche> ne sont pas des actions légales dans le contexte. L'analyse des auteurs est qu'on laisse à l'utilisateur la possibilité de mettre en oeuvre des commandes inutiles et leur solution est d'interdire l'utilisation de ces touches. Le système de règles en devient effectivement beaucoup plus simple, mais l'utilisateur ne dispose plus alors que d'une seule méthode pour assurer les corrections. Le point de vue ergonomique serait plutôt de traiter l'occurrence de ces erreurs comme des indicateurs de l'attente des utilisateurs et donc au contraire d'affecter à ces touches la sémantique de correction attendue.

- Enfin les relations entre les différents critères de qualité proposés par Monk et Dix (1987) ne sont pas très claires. Comme le signalent eux-même les auteurs, le poids des effets imprévisibles dépend des possibilités de récupération des erreurs par annulation des actions antérieures. Il y a donc entre la réversibilité et la prédictibilité des interactions dont une évaluation pertinente doit absolument tenir compte.

## *2.2. Complexité perceptive des affichages*

Une dimension importante de l'utilisabilité d'une interface tient à la qualité de la présentation d'information sur l'écran. De nombreux travaux de recherche se sont attachés à identifier des critères de présentation optimale et les déterminants de la complexité perceptive. Celle-ci est distinguée de la complexité cognitive et peut être analysée de manière indépendante de toute utilisation avant la conception. Les principaux travaux dans ce domaine sont présentés rapidement.

### 2.2.1. Modèle de Tullis (1988)

Tullis (1984) a introduit l'idée d'une analyse automatique d'écrans alphanumériques de telle sorte qu'un programme puisse en caractériser la qualité du point de vue de l'utilisateur. Pour lui, la qualité d'un affichage se manifeste par la réduction du temps moyen de recherche d'un item sur un écran et par une évaluation qualitative positive de l'organisation de la présentation.

#### a) Critères de complexité perceptive

A partir d'une analyse de la littérature dédiée à la présentation d'information (guides ergonomiques et études empiriques)[1], Tullis a isolé un ensemble de critères présentés comme des déterminants de la facilité de lecture. Un programme d'analyse automatique est élaboré pour mesurer les 6 caractéristiques identifiées qui concernent:

- la densité globale d'information = nombre de caractères affichés, exprimé comme un pourcentage de l'espace total d'affichage disponible;

- la densité locale = pourcentage moyen d'occupation dans un angle de 5 degrés autour de chaque caractère;

- le nombre de groupes, i.e. le nombre distincts de groupes de caractères isolables par une métrique de proximité dérivée des concepts de la théorie de la gestalt.

- la taille moyenne des groupes, mesurée par l'angle visuel moyen sous-tendu par le groupe de caractères.

---

[1] Cette analyse est présentée dans Tullis (1983)

- le nombre d'items, i.e. le nombre de labels individuels ou de données sur le dispositif (proche de la densité globale)

- la complexité d'affichage = mesure d'alignement, une moyenne d'incertitude de la position horizontale et verticale de chaque item calculée par rapport à la théorie de l'information

b) Validation de la pertinence des critères

La vérification de la pertinence de ces critères de facilité de lecture d'un écran est assurée par une étude expérimentale conduite auprès de 10 sujets. Chaque sujet voit 520 écrans (26 formats de présentation d'information de trafic aérien et de listes hôtelières), et doit répondre à des questions très précises nécessitant la lecture d'un item affiché. Le temps de recherche est enregistré et les sujets évaluent la facilité de lecture de l'écran sur une échelle en 5 points. Les résultats sont analysés par des corrélations entre le temps de réponse et l'évaluation sur chacune des dimensions. Les diverses transformations statistiques effectuées par Tullis l'autorisent à conclure que:

- les deux paramètres les plus importants pour la détermination du temps de recherche sont ceux qui concernent les groupements de caractères (nombre et taille)
- les deux paramètres les plus importants pour la détermination des évaluations subjectives sont la densité locale et la complexité perceptive;
- la présentation qui optimise le temps de recherche n'est pas nécessairement celle qui optimise les évaluations subjectives et inversement.

Un programme d'analyse automatique des écrans (Display Analysis Program) a été utilisé pour évaluer des écrans ayant par ailleurs fait l'objet d'études approfondies. Le programme est en mesure de prédire avec un degré de précision important le temps de recherche des items et les évaluations subjectives. Il est actuellement utilisé comme outil de développement itératif: le programme analyse les premières versions d'écrans et en propose des aménagements.

c) Problèmes et limites

La prédiction de la qualité des affichages doit être vue comme un complément aux études empiriques et non comme une substitution. Une des limites de l'approche est qu'elle ne s'applique qu'à des alternatives de présentation du même ensemble d'information: il n'est pas possible d'établir une mesure globale de qualité d'affichage d'un dispositif, car il peut comporter des écrans très différents.

De plus, Perlman (1987) signale quelques lacunes qui limitent l'intérêt du modèle pour évaluer un système opérationnel: l'analyse ne prend pas en considération les méthodes usuelles de différenciation des informations et de focalisation de l'attention de l'utilisateur (par exemple la surbrillance, l'encadrement ou l'organisation hiérarchique des informations). Les traitements statistiques utilisées (analyses de régression) n'apportent aucun diagnostic de la qualité et font disparaître certaines informations du fait des pondérations de moyenne.

### 2.2.2. Modèle de Streveler et Wasserman (1985)

Streveler et Wasserman (1985) ont proposé un autre ensemble de mesures quantitatives caractérisant les propriétés spatiales des écrans alphanumériques monochromatiques. Leur objectif est d'opérationaliser les recommandations de conception jugées beaucoup trop floues et de rendre précisément compte de la syntaxe de l'écran (format de présentation des informations, disposition relative des items...). Trois techniques principales sont utilisées: une analyse des regroupements ("boxing analysis"), une analyse de la densité ("hot spot analysis"), et une analyse des alignements[1].

[1]Les algorithmes utilisés pour ces différents indices sont commentés par Tullis (1988) qui les compare à ses propres travaux.

a) Analyse des regroupements

Cette technique définit des groupes d'items à partir de leur proximité. Un "groupe" est un ensemble d'items complètement cerné d'espaces blancs autour duquel on dessine une" boîte", i.e. le plus petit encadrement rectangulaire possible. Ces groupements sont identifiés à partir de principes d'organisation perceptive dérivés de la Gestalt (principe de proximité: des éléments visuels faiblement espacés entre eux et plus éloignés d'autres éléments sont perçus comme appartenant à la même structure, principe de clôture...).

b) Analyse de densité

L'analyse de la densité représente les "intensités" relatives dans un champ visuel comme pourrait le faire des mesures établies par un dispositif optique. Elle est calculée en affectant une valeur binaire à chaque position (0 ou 1 selon qu'elle est libre ou occupée) et en pondérant l'intensité de chaque caractère sur l'écran en fonction du voisinage; les résultats sont présentés comme une carte topographique faisant apparaître les zones foncées de forte densité de caractère. L'hypothèse est que l'oeil est attiré par ces régions lors de la recherche d'information: les pics de densité indiquent les points de focalisation du regard.

c) Analyse des alignements

Cette technique permet de juger de la structure tabulaire d'un écran. Un point d'alignement est calculé par rapport aux extrémités des chaînes de caractères et les auteurs proposent des indices de "force de l'alignement" calculés par rapport au nombre de points d'alignement dans chaque colonne d'information.

Ces trois analyses définissent pour les auteurs des propriétés "de base" des écrans dont ils recherchent ensuite des moyens d' établir les valeurs optimales. Ils envisagent tout d'abord plusieurs mesures quantitatives de charge de l'écran en fonction des unités d'analyse utilisées: par exemple, des comptabilisations établies au niveau des pixels différencient les points et les lettres et rendent compte du rapport texte/ponctuation, la comptabilisation du nombre de groupements est selon eux un autre indicateur intéressant du nombre d'entités composites perçues... D'autres mesures relatives sont proposées:

- distance moyenne entre groupes
- distances moyennes entre groupes par rapport au nombre total de groupes
- taille moyenne des groupes
- nombre de point de focalisation pour analyser l'organisation de l'écran
- nombre total d'alignements répétés
- nombre maximum d'alignement dans une colonne.....

Ces mesures quantitatives sont complétées par des évaluations de l'"esthétique" de l'écran: l'équilibre est défini par la différence absolue entre le centre de masse et le centre physique de l'écran; la symétrie dépend du nombre d'axes de symétrie et la similarité de la taille des boîtes.... Ces mesures ne sont que des possibilités envisagées par Streveler et Wasserman (1985); elles n'ont pas été validées mais des expérimentations validant ces indices de complexité perçue sont envisagées.

d) Problèmes et limites

Cette approche pose un problème identique à celle de Tullis: elle ne pourrait être satisfaisante que dans la mesure où les affichages d'un système sont homogènes, faute de quoi une évaluation de la qualité reste limitée à chacun des écrans. Les attitudes des individus vis-à-vis des dispositifs qu'ils utilisent sont des réactions de synthèse, décrivant une perception globale. La cohérence des formats de présentation d'un écran à l'autre devient alors une dimension essentielle de la complexité perceptive qui n'est pas prise en compte par cette analyse formelle. Les mesures ne considèrent que des aspects statiques de l'écran, or il est fréquent dans les interfaces alpha-numériques que les difficultés soient liées à des variations brutales de densité

d'information. Il faudrait introduire des mesures de contraste. Pour conclure, si les indices de complexité perceptive contribuent à l'évaluation de l'utilisabilité, ils ne peuvent être suffisants pour établir la qualité d'un affichage.

## 2.3. *Génération automatique d'affichages*

### 2.3.1. Approche axiomatique de la qualité des affichages

Selon Perlman (1987) tout affichage doit rendre apparente la structure sous-jacente de l'information. Ce point de vue revient à considérer que la complexité d'un visuel (display) dépend des opérations mises en jeu par celui qui traite l'information. Sur cette base, l'auteur propose un modèle implémentable permettant d'inférer quels attributs d'affichage doivent être utilisés pour mettre en évidence la structure logique d'un ensemble déterminé d'information. Le modèle repose sur une description explicite de cette structure logique et opère à un haut niveau d'abstraction: il met en correspondance une représentation abstraite de la structure de l'information et des méthodes abstraites d'affichage; c'est uniquement lorsque des détails structuraux concrets sont fournis que la valeur exacte des attributs graphiques de présentation est déterminée.

Les relations structurelles entre les entités d'un ensemble d'information sont formalisées dans un réseau sémantique. L'exemple fournit par Perlman est celui d'un formulaire ayant la structure suivante:

ETAT CIVIL
Nom: _____ (nom, Prénom)
Sexe: _ (M ou F) Date de naissance: --/--/-- (jj/mm/aa)
N°SS: -/--/--/--/---/--- Ville:_____

COORDONNEES PERSONNELLES
Adresse: _____ Apt: _____
Ville: _____ Dépt: -- Code Postal: --
Téléphone: (-- -) --/--/--/--

COORDONNEES PROFESSIONNELLES
Titre: _____ Dept: _____
Organisme: _____
Adresse: _____
Ville: _____ Dépt: -- Code Postal: --
Téléphone: (-- -) --/--/--/-- Poste: _____
Email: _____

*Fig. 20: Exemple de formulaire analysé par Perlman (1987)*

La structure logique de cet objet est fournie sous une forme indépendante des dispositifs techniques à partir de 4 relations principales (inclusion, typage, séquentialité, héritage), soit:

| | |
|---|---|
| X in Y | l'item X fait partie de Y |
| X isa Y | X est une instance de Y |
| X after Y | X doit suivre Y |
| X is P | X a la propriété P |

Voici quelques uns des prédicats décrivant le formulaire:

(ETAT_CIVIL COORDONNEES_PERSONNELLES COORDONNEES_PROFESSIONNELLES in FORMULAIRE)
(il y a trois sections dans le formulaire)

(Nom Sexe Date_de_naissance N°SS Ville in ETAT_CIVIL)
(il y a cinq composants dans la section ETAT_CIVIL)
...

Les champs du formulaire sont composés de libellés et de zones d'édition présentés dans cet ordre, soit:

(libellé zone_édition in champ)
(zone_édition après libellé ) ...

L'exploitation de cette description nécessite des règles associant des attributs graphiques aux types de relations identifiées. Ces règles constituent une axiomatique de la présentation d'information organisées selon deux classes principales:

- les *axiomes de structure* définissent la sémantique des relations décrivant l'objet structuré et supportent les inférences basées sur la structure logique sans référence aux affichages. Perlman propose des règles de transitivité et d'héritage de propriétés qui établissent les ressemblances et les différences entre les entités . Par exemple, l'axiome:

(X isa Y) & (Y is P) -> (X is P)

établit que si X est une instance de Y et que Y a la propriété P, X a aussi la propriété P. D'autres règles concernent le contrôle lors du raisonnement et sont destinées à éviter les boucles lors de la décomposition récursive.

- les *axiomes de présentation* établissent la correspondance entre les propriétés structurelles (ressemblances et différences) et les propriétés des affichages. Ces règles exploitent des caractéristiques bien connues dérivées des travaux de la Gestalt et définissent les conditions de séparation, de regroupement, de discrimination.... des objets affichés. Par exemple:

(A in X) & (B in X) & ¬ (C in X)-> D(A,B)< (D(A,C) + D(B,C))

cette règle établit que deux items d'un même groupe doivent être rapprochés les uns des autres mais éloignés des items d'autres groupes.

L'aspect formel intéressant est que des altérations des axiomes d'affichage peuvent modifier radicalement la présentation d'information tout en préservant la structure logique sous-jacente.

Un prototype dans lequel le réseau sémantique est implémenté en LISP a été développé sous UNIX. Le modèle Lisp génère un ensemble de règles qui sont les entrées d'un interpréteur Prolog produisant les attributs de présentation des items. Un module d'évaluation repère l'absence d'attributs différentiels lorsque les objets doivent être distingués et l'absence d'attributs communs lorsque les objets sont similaires. Le développement d'un système expert en conception d'écran est envisagé.

### 2.3.2. Génération automatique des présentations graphiques

Mackinlay (1986) cherche à automatiser la présentation d'informations relationnelles (histogrammes, graphes connectés, dispersion) et veut développer un outil de présentation graphique indépendant d'une application. Deux exigences principales sont recherchées:

- un formalisme doit permettre la codification de critères de conception graphique sous une forme exploitable par un outil de présentation
- le dispositif doit pouvoir générer une large gamme de représentations graphiques pour être capable de traiter des informations très diverses.

Dans la perspective de l'auteur, les présentations graphiques sont des phrases de *langages graphiques* similaires à d'autres langages formels ayant des propriétés syntaxiques et sémantiques bien définies. Les questions clés de conception graphique doivent alors être exprimées en termes de capacité d'expression et d'efficacité de ces langages. La capacité d'expression détermine dans quelle mesure le langage considéré exprime toute l'information requise et uniquement celle-ci; l'efficacité indique dans quelle mesure le langage exploite les capacités du medium d'affichage et correspond aux contraintes du système visuel humain. A titre d'illustration, il est parfois difficile de représenter judicieusement toute l'information dans un système de coordonnées cartésiennes: des chevauchements de labels rendent l'information illisible. D'autres représentations (par histogramme par exemple) sont alors plus adaptées.

L'auteur développe un formalisme comportant un ensemble réduit de primitives graphiques et des opérateurs de composition. A partir de cette algèbre il devient possible de générer une grande variété de représentation. Des techniques d'IA sont utilisées pour implémenter un prototype (APT) basé sur l'algèbre et sur les propriétés formelles du langage graphique.

Ce travail est intéressant dans une perspective d'évaluation ergonomique car d'une part il intègre certaines contraintes de traitement de l'information visuelle par le sujet humain et que d'autre part il rend explicite les conventions de lecture de l'information graphique.

- Contraintes perceptives: l'analyse de l'efficacité d'un langage graphique pose la question de l'évaluation comparative d'un mode de représentation par rapport à un autre. Mackinlay considère que cette question nécessite de prendre en compte les capacités perceptives de l'individu et il propose un ensemble de conjectures relatives à la validité des attributs graphiques d'un objet (le codage) en fonction du type de données représentées. En bref, selon que les ensembles d'informations présentées sont de nature quantitative, ordinale ou nominale, les attributs utilisés sont plus ou moins adéquats. Par exemple le codage des relations par positionnement des objets est également valide dans tous les cas de figure, mais la densité, la saturation, les nuances de couleurs sont traités plus facilement dans le cas de variables ordinales qu'avec les variables discrètes. Ces conjectures n'ont pas été validées mais reposent en partie sur des données expérimentales; elles ont le mérite de hiérarchiser les modalités de codage en fonction de critères objectifs.

- Conventions graphiques: toute communication repose sur le partage de conventions qui restent souvent implicites. Le formalisme utilisé les explicite de façon très précise. Par exemple, si l'on considère le dessin suivant:

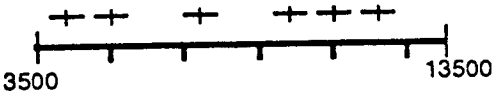



*Fig. 21: Exemple d'objet graphique modélisé par Mackinlay (1986)*

Le langage graphique élémentaire décrivant le positionnement des marques sur l'axe horizontal suppose que chacune d'elles renvoit à une valeur unique des variables du domaine représenté. L'analyse formelle permet de plus d'identifier les erreurs liées à la présentation d'informations additionnelles non représentées dans le domaine initial et qui conduisent à des difficultés d'interprétation.

# CONCLUSION

L'analyse bibliographique présentée ci-dessus a été organisée en distinguant *4 contextes d'évaluation* auxquels est régulièrement confronté l'ergonome: le diagnostic d'usage de systèmes existants, les tests réalisés en cours de conception, les évaluations comparatives de logiciels verticaux et le contrôle a priori de la qualité de l'interface. Des techniques mises en oeuvre dans chacun de ces contextes ont été discutées en contrastant les *approches empiriques* (centrées sur le recueil de performances d'utilisation) et les *approches analytiques* (centrées sur la modélisation de l'interface homme-machine).

A l'issue de cette analyse, la question de savoir comment évaluer d'une façon systématique la qualité ergonomique d'une IHM reste ouverte: aucun des travaux présentés ne répond de façon réellement satisfaisante et n'est en mesure d'établir un diagnostic complet d'une interface utilisateur car ils présentent tous des limites importantes. La plupart d'entre eux ne concerne qu'un aspect de l'interface: par exemple, ils portent de façon exclusive sur le langage d'entrée (langage de commande) ou sur le langage de sortie (affichages) et ne prennent pas en considération leurs relations pour établir la qualité du *dialogue* homme-machine. Pour mettre en évidence de façon synthétique les lacunes des différents travaux mentionnés, on les résume ici en montrant qu'ils s'adressent à deux niveaux d'évaluation d'une interface, l'ergonomie étant recherchée au niveau des *propriétés intrinsèques* de l'interface ou de son adéquation à des *critères externes*.

## 1. Deux niveaux d'évaluation d'une interface

### *1.1. Evaluation des propriétés intrinsèques de l'interface.*

#### 1.1.1. Caractérisation du langage d'entrée

a) Structure du langage de commande

Il n'existe pas actuellement de formalisme caractérisant la qualité ergonomique d'un langage d'entrée de façon satisfaisante. Les métriques proposées dans le premier modèle de Reisner (1981) (taille du lexique, nombre de règles et redondance) sont trop simplistes pour représenter la complexité d'un langage de commande. Celles proposées par Monk et Dix (1987) dans le modèle "Black Box" sont encore mal définies et doivent être opérationalisées; de plus, elles demandent à être validées. Cependant, ce modèle décrit les changements d'état de l'interface du point de vue de l'utilisateur en les organisant en contextes de travail. Cette unité d'analyse reste proche de la situation d'interaction telle que la voit le sujet et l'étude des difficultés d'utilisation potentielles dans chaque contexte devient possible, mais le problème se pose ensuite d'en dériver une évaluation globale du logiciel.

L'utilisation de métriques dérivées de "CLG" pour déterminer la complexité de l'interface (structure des tâches et longueur des méthodes syntaxiques, Sharatt, 1987) suppose une description dans le formalisme CLG et ces métriques n'ont pas été validées.

Par contre, les guides de conception regorgent de recommandations, de règles et de normes établissant de façon très locale les exigences du système de commande (cf. les propriétés des menus: dimensions, nombre d'item maximum, niveau de profondeur....). La liste de ces exigences définit un ensemble de critères d'évaluation du système de commande.

b) Cohérence des procédures

La typologie des niveaux de cohérence introduite par Kellog (1987) enrichit les dimensions d'évaluation. Le travail expérimental présenté ne concerne cependant que la cohérence sémantique (régularité des méthodes sémantiques) et traite uniquement de l'accès aux fonctions. Il est difficile d'en dériver des indicateurs précis décrivant de façon formelle cette notion.

L'évaluation systématique de la cohérence d'une IHM supposerait des méthodes spécifiques pour chacun des aspects de cohérence interne et externe: cohérence conceptuelle, sémantique, syntaxique et lexicale. Les travaux expérimentaux qui ont permis d'établir les recommandations ergonomiques identifient quelques-uns des critères de cohérence. Par exemple la cohérence du lexique du commande tient à la préservation des relations hiérarchiques des labels (sous-ordonnés et super-ordonnés) dans les emboîtements de menu. L'importance relative des différents types de cohérence n'est pas connue.

### 1.1.2. Caractérisation du langage de sortie

#### a) Syntaxe de l'écran

Les travaux de Tullis (1988) et de Streveler et Wasserman (1985) répondent partiellement aux questions de qualité des affichages. Ces analyses permettent d'avoir pour chaque écran une estimation du temps de recherche des items et de l'appréciation subjective de l'utilisateur. Le fait qu'elles ne concernent que des écrans indépendants et que les critères d'évaluation ne prennent pas en compte les méthodes d'affichage particulières (surbrillance, soulignement, clignotement...) constitue une limitation importante pour l'évaluation d'une IHM en situation naturelle. De plus, ces modèles ne peuvent pas non plus être utilisés pour comparer des logiciels. Il est fréquent dans les interfaces alpha-numériques que des variations brutales de densité d'information soient à l'origine de difficultés d'utilisation. Des mesures de transition d'écran (variation du contraste, de la densité...) devraient permettre de développer des évaluations plus réalistes.

Ces modèles de complexité perceptive doivent être vus comme des aides à l'évaluation, i.e. des outils de confirmation et de validation permettant d'exprimer sous une forme quantitative les impressions subjectives que l'on peut avoir devant un écran.

#### b) Cohérence du langage de sortie

On retrouve à ce niveau différents types de cohérence: la cohérence syntaxique du langage de sortie se traduit par exemple par la régularité des positions d'affichage des mêmes éléments sur les différents écrans. La cohérence lexicale apparait à travers l'invariance des codes: les modifications des attributs graphiques d'un objet affiché doivent obéir à des règles bien définies. D'autres aspects concernent la structure des messages d'erreurs, des messages de guidage et la cohérence externe des codes utilisés pour différencier les objets affichés et leurs états.

C'est certainement à ce niveau qu'une évaluation automatique peut être réalisée le plus facilement: la littérature ergonomique propose un ensemble de règles concernant directement ces aspects. Elles sont plus ou moins bien structurées mais ne sont pas ambiguës.

Incidemment, les travaux de Kellog (1987) suggèrent que l'évaluation de la cohérence d'une interface peut être assurée "en incidence" à partir de la mise en évidence de ses effets: étant donné qu'elle améliore la capacité de rappel et d'inférence du sujet, sous réserve d'un contrôle expérimental rigoureux, ce type de tâche permet de différencier les interfaces.

## *1.2. Adéquation de l'interface*

Ce deuxième niveau d'évaluation cherche à déterminer dans quelle mesure l'interface est satisfaisante du point de vue des tâches qui doivent être assurées et/ou du point de vue des activités de l'utilisateur.

### 1.2.1. Adéquation aux tâches

Les deux méthodologies d'évaluation comparative qui ont été présentées de façon assez détaillée ci-dessus permettent de répondre à cette question moyennant des adaptations au domaine d'application. Les critères utilisés pour déterminer la capacité fonctionnelle des logiciels sont des indicateurs de l'adéquation aux tâches. Les évaluations qui en résultent sont cependant

biaisées du fait qu'elles concernent uniquement les performances sans erreur: elles donnent alors une estimation approximative de l'utilisation réelle. Une évaluation portant sur l'adéquation aux tâches mises en oeuvre par les utilisateurs suppose de définir le niveau d'analyse de ces tâches. Ce peut être uniquement une référence fonctionnelle comme dans les bancs d'essai de tâche, i.e. sans relation avec la structure de l'activité des opérateurs ou bien une description résultant d'observations en situation naturelle. L'évaluation dans ce cas a plus de chance d'avoir une validité écologique.

### 1.2.2. Adéquation du langage d'entrée

#### a) Cohérence externe et représentations mentales

Dans quelle mesure le système de commande est-il bien adapté à la façon dont les utilisateurs se représentent le dispositif ? Cette question est importante puisqu'il peut en découler de nombreuses confusions de commande. Dans les travaux présentés ci-dessus, l'évaluation d'un logiciel sur cette dimension passe par l'identification des "modèles mentaux" des utilisateurs mais les réserves à l'égard des techniques utilisées sont justifiées. Le véritable problème est qu'une seule tâche est utilisée pour inférer les "représentations mentales" ce qui laisse supposer qu'une trop grande confiance est accordée aux techniques statistiques de classification hiérarchique. On peut penser en effet que l'inférence de la structure d'une représentation mentale circonstancielle ou d'une organisation conceptuelle plus stable à partir d'une seule expérimentation est une opération risquée. Ces inférences doivent être validées par d'autres tâches confirmant que les groupements des commandes réalisés au cours des tâches de classification traduisent bien un modèle mental du dispositif[1].

Ces études de cohérence externe relèvent encore de travaux de recherche dont il parait a priori difficile de dériver une batterie de tests bien structurée permettant d'évaluer sans ambiguïté la cohérence d'une application.

#### b) Adéquation lexicale

Selon les distinctions introduites par Kellog, une dimension de la cohérence externe tient à la congruence du lexique de commande avec la langue naturelle. Un aspect de cette adéquation concerne par exemple l'utilisation de la terminologie en vigueur dans un système de travail lors de la définition du lexique de commande du dispositif. L'idée est que l'utilisation peut être rendue d'autant plus facile que les opérateurs retrouvent au niveau de l'interface les termes qu'ils emploient. Une évaluation sur cette dimension pourrait consister en une comparaison de deux listes, l'une décrivant le "langage opératif" (i.e. le lexique technique utilisé dans le travail) et l'autre le lexique de commande. De toute évidence, cette comparaison ne serait pas suffisante pour qualifier l'interface: une cohérence de surface ne garantit pas la facilité d'utilisation mais peut au contraire, selon la sémantique des commandes être une source d'ambiguïté importante. La question rejoint le principe de prédictibilité de Monk et Dix (1987) selon lequel les attentes de l'utilisateur à propos de l'effet des commandes doivent être satisfaites.

Ces remarques montrent que s'il est commode de différencier sur le papier des types de cohérence, il peut être plus difficile de les étudier séparément dans la réalité.

### 1.2.3. Adéquation du langage de sortie

Dans quelle mesure les écrans d'une application permettent-ils aux utilisateurs d'atteindre leurs objectifs de travail ? Cette vaste question recouvre un ensemble de dimensions qui n'ont été mentionnées que très partiellement ci-dessus. L'évaluation de la qualité des messages

---

[1] La psychologie expérimentale classique propose un ensemble de paradigmes permettant d'effectuer ces validations croisées (par exemple, mesures de temps de réponse lors de comparaison de paires de commandes plus ou moins proches, évaluation qualitative de groupements plus ou moins proches du modèle mental identifié...). Voir Bewley et al. (1983) pour une illustration plus précise.

d'erreur et des représentations iconiques en sont deux aspects, mais tous les affichages sont concernés: les messages de guidage, les alarmes, les aides, les objets de l'application.

Les évaluations réalisées sur ce point posent essentiellement des problèmes de compréhension qui peuvent être abordés par la spécification d'objectifs de performance, selon les propositions de l'ingéniérie de l'évaluation. Il s'agit de poser par exemple comme exigence de performance que les utilisateurs doivent pouvoir se sortir de toutes les situations d'erreur grâce aux seuls messages de guidage. Les tests mis en oeuvre pour vérifier la satisfaction de cet objectif doivent sans ambiguïté montrer que le sujet a acquis une compréhension de l'information affichée (par exemple en paraphrasant les messages).

Les travaux relatifs à l'axiomatique des affichages (Perlman, 1987 et Mackinlay, 1986) qui ont été rapidement évoqués ci-dessus pourraient constituer de bons outils d'évaluation dans la mesure où ils établissent une relation formelle entre la structure initiale de l'information et des attributs d'affichage. On peut alors espérer qu'ils permettent de définir ce que doit être un écran qui préserve l'information initiale.

Un évaluateur ne peut se contenter d'une évaluation portant de façon exclusive sur l'un ou l'autre de ces niveaux: il doit pouvoir caractériser les points forts et les points faibles d'une IHM de façon transversale et l'analyse doit porter aussi bien sur le langage de commande que sur l'organisation des écrans. La mosaïque de travaux présentée suggère de plus qu'il est vain de vouloir développer un outil capable de fournir une évaluation définitive de n'importe quel logiciel: de toute évidence, cet objet est trop complexe pour qu'on puisse espérer résumer ses qualités et ses défauts par une simple mesure permettant de le positionner sur une échelle de valeur. Un outil vraiment satisfaisant devrait en fait être constitué d'un véritable environnement d'évaluation.

## 2. Assistance à l'évaluation

L'idée d'un environnement d'assistance à l'évaluation dans un contexte de conception peut être illustrée à partir du tableau ci-dessous. Ce tableau, dérivé du modèle proposé dans le projet ESPRIT "HUFIT", distribue les différentes techniques d'évaluation disponibles en considérant un cycle de vie du produit comportant 5 phases: le planning, le développement, l'évaluation, (testing), l'implémentation et l'utilisation.

| Planning | Développement | Evaluation | Implémentation | Utilisation |
|---|---|---|---|---|
| Performances d'usage | sélection d'alternatives | Etudes expérimentales | expertise ⟷ | |
| | | | monitoring ⟷ | |
| Analyse du travail | méthodes formelles<br>- CLG<br>- GOMS<br>- complexité cognitive | Questionnaire | méthodes cliniques ⟷ | |
| Grille d'évaluation | | Prototypage | station d'évaluation | cahier de doléance |

Cette organisation va à l'encontre des idées reçues en matière d'évaluation: elle montre que celle-ci constitue un *processus* parallèle à la conception, qu'elle ne peut être réduite à une simple étape ponctuelle et qu'une *équipe d'évaluation* doit pouvoir utiliser un ensemble d'outils allant des techniques cliniques aux méthodes formelles à chaque étape de la vie d'un produit. La mise en oeuvre d'une approche de ce type reste cependant théorique du fait des contraintes pesant sur la conception. Elle ne pourra être effective que dans la mesure où des outils fiables, systématiques et faciles à utiliser seront disponibles.

Dans l'idéal, un évaluateur devrait disposer d'un environnement à partir duquel il puisse analyser des aspects aussi divers que la qualité d'un système d'aide, l'adéquation fonctionnelle du

logiciel ou que la cohérence des méthodes interactives utilisées. Dans cette perspective, l'automatisation totale du processus paraît difficile: la simple évaluation des messages de guidage supposerait une connaissance du monde réel qu'il est illusoire de vouloir représenter dans une base de connaissances. Cette difficulté peut être contournée en faisant appel aux connaissances et au jugement de l'évaluateur et en mettant en jeu des techniques de questionnement.

Par ailleurs, l'évaluation d'un logiciel nécessite une compréhension de son *contexte d'utilisation*: la connaissance du monde pondère l'importance des propriétés d'une interface en fonction des caractéristiques de la population, des exigences de la tâche, du matériel utilisé... Par exemple, une application peut être jugée satisfaisante parce que les menus qu'elle propose permettent à une population de novices d'acquérir une connaissance de l'utilisation en un temps raisonnable et parce que l'environnement documentaire est facilement utilisable. Ce même logiciel peut être jugé inadéquat si l'on cherche à mettre à la disposition d'expert des outils performants, évolutifs, comportant un langage de macro-programmation permettant l'extension des fonctions. A partir du moment où l'on admet que l'évaluateur doit moduler son point de vue sur les qualités d'une interface en fonction des exigences du contexte, il devient important de lui fournir un environnement lui permettant *d'établir* et de *gérer ces compromis*. A titre d'exemple, à l'issue d'une évaluation comparative, l'évaluateur peut être placé dans une situation de choix entre deux dispositifs dont l'un distribue l'information requise pour une tâche donnée sur des écrans successifs d'une qualité "optimale" et l'autre qui présente l'ensemble de ces informations sur un seul écran mais sous une forme mal structurée. La préférence résulte d'un compromis entre la complexité perceptive et les difficultés d'accès aux informations; la pondération de l'importance relative de ces difficultés n'est pas une opération triviale. L'issue d'un diagnostic ergonomique doit faire l'objet d'une interprétation contextuelle et l'utilisateur doit être informé des facteurs pertinents et comprendre leur portée.

Enfin, les experts en ergonomie des interfaces utilisateur produisent des évaluations qui améliorent considérablement la qualité des interfaces sans nécessairement recourir à des outils analytiques ou procéder à des évaluations empiriques systématiques. Le processus d'évaluation qu'ils mettent en oeuvre est réalisé en deux temps: ils repèrent rapidement quelques aspects particulièrement critiques du dispositif et étudient éventuellement ensuite par des analyses plus structurées les difficultés liées à ces points faibles.

Un environnement d'évaluation devrait assister ces deux étapes. Dans l'idéal, un pré-diagnostic établi à partir d'une description partielle de l'interface devrait en caractériser les propriétés intrinsèques en faisant apparaître ses principaux points critiques (caractéristiques du langage de commande, organisation des menus, indices de complexité perceptive...). Dans un second temps, l'évaluateur doit avoir la possibilité de réaliser des analyses plus précises sur des points particuliers, jugés importants dans le contexte considéré. Tout le problème d'une aide à l'évaluation est d'estimer l'importance des difficultés d'utilisation mises en évidence. On a rapporté ci-dessus des travaux de conception dans lesquels aucun aménagement n'était réalisé malgré la détection de difficultés, soit parce que le coût de l'implémentation était trop important par rapport à l'amélioration de la performance, soit parce qu'un avis d'expert posait que les avantages de la caractéristique considérée étaient à terme supérieurs aux difficultés détectées. La hiérarchisation des problèmes par rapport aux exigences du contexte devient alors une condition de l'utilité d'un outil d'assistance à l'évaluation.

. . . . . .

# BIBLIOGRAPHIE


Anderson, J.R. (1976) Language, Memory and Thought. Hillsdale, New Jersey: LEA.

Apperley, M.D. et Field, G.E. (1985) A comparative evaluation of menu-based interactive human-computer dialogue techniques. In *Human-Computer Interaction* - INTERACT'84, B. Schackel (Ed.) Elsevier Science Publishers B.V.: North-Holland, IFIP, 296-300.

Bailey, W. A. et Kay, E.J. (1987) Structural analysis of verbal data. In *Human Factors in Computing Systems-IV*, J.M. Carroll et P.P. Tanner (Eds.). ACM, North-Holland, Amsterdam, 297-301.

Bannon, L. et O'Malley, C. (1985) Problems in evaluation of human-computer interfaces: a case study. In *Human-Computer Interaction* - INTERACT'84, B. Schackel (Ed.) Elsevier Science Publishers B.V.: North-Holland, IFIP, 709-713.

Bewley, W.L., Roberts, T.L., Schroit, D. et Verplank, V.L. (1983) Human factors testing in the design of Xerox's 8010 "STAR" Office Workstation. In *Human Factors in Computing Systems-I*, A. Janda (Ed.). ACM, North-Holland, Amsterdam, 72-77.

Borenstein, N.S. (1985) The evaluation of text editors: a critical review of the Roberts and Moran methodology based on new experiments. In *Human Factors in Computing Systems-II*, L. Borman et B. Curtis (Eds.), ACM, North-Holland, Amsterdam, 99-105.

Bury, K.F. (1985) The iterative development of usable computer interfaces. In *Human-Computer Interaction*- INTERACT'84, B. Schackel (Ed.) Elsevier Science Publishers B.V.: North-Holland, IFIP, 743-748.

Butler, K. A. (1985) Connecting theory and practice: a case study of achieving usability goals. In *Human Factors in Computing Systems-II*, L. Borman et B. Curtis (Eds.) ACM, North-Holland, Amsterdam, 85-88.

Card, S. K., Moran, T. P., Newell A. (1983) The psychology of human-computer interaction. LEA, Hillsdale, NJ.

Caroll, J.M. et Thomas, J.C. (1982) Metaphor and the cognitive representation of computing systems. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-12 (2), 107-116.

Caroll, J.M. et Rosson, M.B. (1985) Usability specifications as a tool in interactive development. In *Advances in Human-Computer Interaction*, H.R. Hartson (Ed.), Vol.1: Ablex, Norwood, N.J., 1-28.

Cohill, A.M., Gilfoil, D.M. et Pilitsis, J.V. (1988) Mesuring the utility of application software. In *Advances in Human-Computer Interaction*, H.R. Harston et D. Hix (Eds.), Vol. 2: Ablex, Norwood, N.J., 128-158.

Clauer, C.K. (1982) Methodology for testing and improving operators publications. Proc. of Office Automation Conference. American Federation of Information Processing Societies, San Fransico, 867-873.

Eason, K. D. (1984) Towards the experimental study of usability. *Behaviour and Information Technology*, 3, (2), 133-145.

Embley, D.W., Lan, M.T., Leinbaugh, D.W. et Nagy, G. (1978) A procedure for predicting program editor performance from the user's point of view. *Int. J. Man-Mach. Stud.*, 10, (6), 639-650.

Embley, D.W. et Nagy, G. (1981) Behavioral aspects of text editors. *ACM Computing Surveys*, 13, (1), March 1981, 33-70.

Ericsson, K.A. et Simon, H.A. (1984) Protocol Analysis: verbal reports as data. Cambridge, MA: MIT Press.

Fischer, G. et Lemke, A. (1988) Construction kits and design environnements: steps toward human problem-domain communication. In *Human-Computer Interaction*, Vol. 3, LEA, 170-22.

Good, M, Spine, T.M., Whiteside, J. et George, P. (1986) User derived impact analysis as a tool for usability engineering. In *Human Factors in Computing Systems-III*, M. Mantei et P. Orbeton (Eds.), ACM, North-Holland, Amsterdam, 241-246.

Grawitz, M. (1974) Méthode des sciences sociales. Dalloz, Paris.

Hammond, N., Hinton, G., Barnard, P., MacLean, A., Long, J. et Whitefield, A. (1985) Evaluating the interface of a document processor: a comparison of expert judgement and user observation. In *Human-Computer Interaction*- INTERACT'84, B. Schackel (Ed.) Elsevier Science Publishers B.V.: North-Holland, IFIP, 725-729.

Hanson, S.J., Kraut, R.E. et Farber, J.M. (1984) Interface design and multi-variate analysis of UNIX command use. *ACM Transactions on Office Information Systems*, 2, 42-57. Cité dans Teubner et al. (1988)

Hewett, T.T. et Meadow, C.T. (1986) On designing for usability: an application of four key principles. In *Human Factors in Computing Systems-III*, M. Mantei et P. Orbeton (Eds.), ACM, North-Holland, Amsterdam, 247-251.

Isa, B.S., Boyle, J.M., Neal, A.S. et Simons, R.M. (1983) A methodology for objectively evaluating error messages. In *Human Factors in Computing Systems-I*, A. Janda (Ed.), ACM, North-Holland, Amsterdam, 68-71.

Kellog, W.A. (1987) Conceptual consistency in the user interface: effect on user performance. In *Human-computer interaction*- INTERACT'87, H. J. Bulinger et B. Shackel (Ed.), Elsevier Science Publishers B. V., North-Holland, 389-394.

Kellog, W.A. et Breen, T.J. (1983) Evaluating user and system models: applying scaling techniques to problems in Human-Computer Interaction. In *Human Factors in Computing Systems-IV*, J. M. Caroll et P.P. Tanner (Eds.), ACM, North-Holland, Amsterdam, 303-308.

Kieras, D.E. (1988) Toxards a practical GOMS model methodology for user interface design. In M. Helander (Ed.) *Handbook of human-computer interaction.* Elsevier Science Publisher B.V., North-Hommand, 135-157.

Kieras, D.E. et Polson, P.G. (1985) An approach to the formal analysis of user complexity. *Int. J. Man-Mach. Stud.*, 22, 365-394.

Lund, M. a. (1985) Evaluating the user interface: the candid camera approach.In *Human factors in Computing Systems II*, L.Borman et B.Curtis (Eds.) ACM, North-Holland, Amsterdam, 107-113.

McDonald, J.E., Dearholt, D.W., Paap, K.R. et Sohvaneveldt (1986) A formal design methodology based on user knowledge. In *Human Factors in Computing Systems-III*, M. Mantei et P. Orbeton (Eds.), ACM, North-Holland, Amsterdam, 285-290.

Mackinlay, J. (1986) Automating the design of graphical presentation of relationnal information. *ACM Transactions on Graphics*, 5, (2), 110-141.

Monk, A., F. et Dix, A. (1987) Refining early design decisions with a black box model. In *People and Computers III* , D.Diaper et R.Winder (Eds.), Cambridge University Press, 147-158.

Moran, T.P. (1981)The Command Language Grammar: A representation of the user interface of interactive computer systems. *Int. J. Man-Mach. Stud.*, 15, 3-50.

Neal, A.S. et Simon R.M. (1983) Playback: a method for evaluating the usability of softwar and its documentation. In *Human Factors in Computing Systems-I*, A. Janda (Ed.), ACM, North-Holland, Amsterdam, 78-82.

Novara, F., Bertaggia, N. et Allamano, N. (1987a) Usability evaluation and feedback to designers- an experimental study. In *Human-Computer interaction* INTERACT'87, H. J. Bulinger et B. Shackel (Ed.), Elsevier Science Publishers B. V., North-Holland, 337-340.

Novara, F., Bertaggia, N., Dillon, A. et Bonner,J. (1987b) The evaluation of products using the usability methodology with proposals for product development. Working paper A5.3a, ESPRIT Project 385- HUFIT/04-OLI-11/87.

O'Bury, K.F. (1983) Prototyping in CMS: Using prototypes to conduct human factors tests of software during development. IBM Human Factors Center Tech. Rep. HFC-43 (IBM General product Division, 5600 Cottle Road, San Jose, CA 95143) Cité dans Teubner, et Vaske (1988)

Ogden , E.C. et Boyle, J.M. (1982) Evaluating human-computer dialog styles: command versus form/fill-in for report modification. *Proc. of the Human Factors Society* 26th Annual Meeting, Santa Monica, CA, 542-545. Cité dans Teubner, et Vaske (1988)

Palmer, J.E., Duffy, T.M., Gomoll, K., Gomoll, T., Palmquist-Richards, J. et Trumble, J. A. (1987) The design and evaluation of on line help for UNIX EMACS: access mechanisms. In *Human-Computer interaction* INTERACT'87, H. J. Bulinger et B. Shackel (Ed.), Elsevier Science Publishers B. V., North-Holland, 461-466.

Payne, S.J. et Green, T.R.G. (1986) Task-action grammars: a model of the mental representation of task languages. In *Human-Computer Interaction*, 2, 93-133.

Perlman, G. (1985a) Electronic surveys. *Behavior Research: Methods, Instruments and Computers*, 17 (2), 203-205.

Perlman, G. (1985b) Making the right choice with menus. In *Human Factors in Computing Systems-II*, L. Borman et B. Curtis (Eds.), ACM, North-Holland, Amsterdam,

Perlman, G. (1987) An axiomatic model of information presentation. *Proc. of the 1987 Human Factors Society Meeting*. New York: Human Factors Society, 1229-1233.

Perlman, G. (1988) User Interface development. *SEI Curriculum*, Module SEI-CM-17-1.0 Carnegie Mellon University, MIT.

Polson, P.G. (1987) A quantitative model of human-computer interaction. In *Interfacing thought: cognitive aspects of human-computer interaction*, J.M. Carroll (Ed.). Cambridge, MA: Bradford, MIT Press.

Reisner, P (1981) Formal grammar and human factors design of an interactive graphics System. *IEEE Tansactions on Software Engineering*, SE.7, (2), March.

Reisner, P. (1984) Formal Grammar as a tool for analyzing ease of use; some fundamental concepts. In *Human factors in Computer Systems*. J.C.Thomas, M.L. Schneider (Eds.) Ablex, New Jersey.

Roberts, T.L. et Moran, T.P. (1983) The evaluation of text editors: methodology and empirical results. *Com. of the ACM*, 26, (4), April, 265-283.

Root, R.W. et Draper, S. (1983) Questionnaire as a software evaluation tool. In *Human Factors in Computing Systems-I*, A. Janda (Ed.), ACM, North-Holland, Amsterdam, 83-87.

Rushinek, A. et Rushinek, S. (1986) What make users happy ? *Com. of the ACM*, july 1986, 29, (7), 594- 598.

Senach, B. et Alengry, P. (1985) Evaluation d'un dispositif d'assistance à la conduite et au dépannage de chaînes de fabrication automatisées. *Rapport Technique INRIA*, février 85.

Senach, B. et Pichancourt, I. (1986) Assistance informatisée au diagnostic de pannes en milieu industriel: 1. Définition des fonctionnalités du dispositif par la technique des incidents critiques. *Rapport Technique INRIA*.

Shackel; B. (1984) The concept of usability. In *Visual Display Terminals*, J. Bennett, D. Case, J. Sandelin et M. Smith (Eds.), Prentice hall, Englewood Cliffs, N.J., 45-88.

Sharatt , B. (1987) The incorporation of earlY interface evaluation into Command language Grammar specification. In *People and Computers III*, D.Diaper et R.Winder (Eds.), Cambridge university Press: Cambridge, 11-28.

Shneiderman, B. (1987) Designing the user interface: strategies for effective human computer interaction. *Readings*, MA: Addison-Wesley.

Smith, D. C., Irby, C, Kimball, R., Verplank., B. (1982) Designing the STAR user Interface. *Byte*, 7, (4), 242-282.

Smith, S.L. et Mosier, J.N. (1984) A design evaluation checklist for user-system interface software. Report # MTR-9480 EDS_TR_84-358.The MITRE Corporation, Bedford, MA.

Streveler, D. J. et Wasserman, A. I. (1985) Quantitative measures of the spatial properties of screen design. In *Human Computer Interaction:* INTERACT '84, B. Shackel (Ed.) Amsterdam: North-Holland, 81-89.

Sweeney. M. et Dillon. A. (1987) Methodologies employed in the psychological evaluation of H.C.I. In *Human-Computer Interaction-* INTERACT'87. H.J. Bullinger and B. Shackel (Eds.) Elsevier Science Publishers, North-Holland, Amsterdam, 367-373.

Teubner, A.L. et Vaske, J.J. (1988) Monitoring computer users' behaviour in office environnements. *Behaviour and Information Technology*, 7, (1), 67-78.

Thomas, J.C. et Kellog, W.A. (1989) Minimizing ecological gaps in interface design. *IEEE Software*, January 1989, 78-86.

Tullis, T. S. (1983) The formatting of alphanumeric displays: a review and analysis. *Human Factors*, 25 (6), 657-682.

Tullis, T.S. (1984) A computer based tool for evaluating alphanumeric displays. In *Human Computer Interaction*: INTERACT '84, B.Shackel (Ed.) Amsterdam: North-Holland, 719-723.

Tullis, T.S. (1985) Designing a menu-based interface to an operating system. *Proc. of CHI*, ACM, North-Holland, Amsterdam, 79-84.

Tullis, T.S. (1988) A system for evaluating screen formats: research and application. In *Advances in Human-Computer Interaction*, H.R. Harston et D. Hix (Eds.), Vol. 2: Ablex, Norwood, N.J., 214-286.

Tytyk, E.S. (1988) Requirements and the method of hierarchization and selection of ergonomic check-list items. In *Man-Machine Systems: analysis, design and evaluation.* Proc. of the IFAC/IFIP/IEA/IFORS Conference, Vol.1; Oulu, Finland, 14-16 June 1988, 244-246.

Waldhör, K. (1989) Creating and using advanced user interfaces using a knowledge based approach. In *Designing and using Human-Computer Interfaces and Knowledge based Systems*, G. Salvendy et M.S. Smith (Eds.), Elsevier Science Publisher (Amsterdam), 869-876.

# FIGURES ET TABLEAUX




Imprimé en France
par
l'Institut National de Recherche en Informatique et en Automatique